

LAMPIRAN A
DATA MATA KULIAH

LAMPIRAN A**Data Mata Kuliah**

Kode Mata Kuliah	Nama Mata Kuliah	sks	Kotak	Semester
000000001	Agama Katolik	2	1	2
000000002	Agama Non Katolik	2	1	2
000000006	Etika Sosial	2	1	4
530316306	Perancangan TLP	3	2	6
530314322	OR II (A/B)	3	2	4
530306341	MPI	2	1	5
000000014	Kewirausahaan	2	1	6
53031632	SISPRO	2	1	6
530303321	OR I	3	2	3
530317308	FMS	2	1	6
530306352	CIM	3	2	7
530302212	Proses Produksi I	3	2	2
530304226	TTL (A/B)	2	1	4
530312232	Praktikum Fisika (A/B/C)	1	2	2
530301131	Fisika II	2	1	2
530304329	Ekonomi Teknik (A/B)	3	2	4
530305338	Analisa Keputusan	2	1	5
530317340	AKP	1	1	7
530305336	Pengendalian Kualitas	2	1	5
530317350	Manajemen Logistik	3	2	8
530317345	Praktikum AKP	1	2	7
530302216	Pengetahuan Bahan (A/B)	3	2	2
530307362	Pemodelan Sistem	3	2	6
530314317	Teknik Peramalan (A/B)	2	1	4
530316371	Manajemen Kualitas	2	1	8
530302215	Mekanika Teknik	3	2	2
530303323	MSDM (A/B)	2	1	2
530316337	Praktikum TLP (A/B)	2	3	6

530305335	Sistem Informasi Manajemen	2	1	7
530317361	Decision Support System	2	1	8
530305333	Statistik Industri II (A/B)	2	1	4
530304332	Statistik Industri I	3	2	3
530306344	Desain Eksperimen	2	1	5
530301201	Kalkulus I	3	2	1
530314302	APK (A/B)	3	2	4
530317379	Teknik Peramalan Lanjutan	3	2	8
530306347	Ergonomi Industri	2	1	5
530303325	Psikologi Industri (A/B)	2	1	2
530314326	Prakt. T. Peramalan (A/B/C/D)	1	2	4
530305334	Prakt. Statistik Industri (A/B/C)	1	2	4
530312234	Kimia II	2	1	2
530316307	Praktikum Pemodelan (A/B)	1	2	6

LAMPIRAN B
DATA DOSEN

LAMPIRAN B**Data Dosen**

Kode Dosen	Nama Dosen	Mata Kuliah Yang Diajar
	<u>Tetap :</u>	
1	Harry Tjahjono, ST	Perancangan TLP OR II (A/B) MPI Kewirausahaan
2	Dian Retno Sari Dewi, ST., MT	SISPRO OR I FMS Prakt.T.Peramalan (B/C) Prakt Statistik Industri (A)
3	Kwa See Yong, ST., MT	CIM Proses Produksi I TTL (A/B) Praktikum Fisika (C) Fisika II
4	Martinus Edy Sianto, ST., MT	Ekonomi Teknik (A/B) Analisa Keputusan APK (A/B)
6	IG. Joko Mulyono, STP., MT	AKP Pengendalian Kualitas Manajemen Logistik Praktikum AKP Prakt. T.Pemodelan (A/B)
8	Hadi Santosa, Ir., MM	Praktikum Fisika (A/B) Pengetahuan Bahan (B)
9	Hendry Rahardjo, ST	Pemodelan Sistem Teknik Peramalan (A/B) Manajemen Kualitas

10	Bambang Widjanarko, Ing., M.Eng	Prakt. T. Peramalan (A/B) Prakt Statistik Industri (A) Mekanika Teknik MSDM (B) Pengetahuan Bahan (A)
24	Dini Endah, ST	Prakt. T. Pemodelan (A) Prakt. T. Peramalan (C)
	<u>Khusus :</u>	
11	Anastasia Lydia, ST., M.Sc., MMT	Praktikum TLP (A/B) SIM Decision Support System
12	Prof. Susanti L., M.Stat., Ph.D	Statistik Industri II (A/B) Statistik Industri I Desain Eksperimen
13	I Nyoman Budiantara., Dr., MS., Drs	Kalkulus I
14	Vincent Winarto, Ir., MSc	MSDM (A)
15	ENGR., Maria Socorro, BSIE, MEM	APK (A/B)
16	Suhartono, S.Si., M.Sc	T. Peramalan (A/B) Teknik Peramalan Lanjutan
17	Ratna Dewi, ST., MT	Prakt. T. Pemodelan (B)
19	Trifosa K. Novita, ST	Praktikum TLP (B) Ergonomi Industri
20	Rudy Susanto, ST	Psikologi Industri (A/B)
21	Chalas Adi Jaya, ST	Prakt. T. Peramalan (D) Prakt Stat Industri (B/C)
22	Soni Hartanto, ST	Prakt Statistik Industri (C)
25	Edwi Nugrohadi, SS	Kimia II
26	Drs. Suyono A, S.Th	Agama Katolik (E) Agama Non Katolik (E) Etika Sosial (E)

Mata Kuliah yang diajar 2 dosen :

Kode Dosen	Mata Kuliah Yang Diajar
4,15	APK (A/B)
6, 24	Praktikum Teknik Pemodelan (A)
6, 17	Praktikum Teknik Pemodelan (B)
2,9	Praktikum Teknik Peramalan (B)
2,24	Praktikum Teknik Peramalan (C)
9, 16	Praktikum Teknik Peramalan (A)
21,22	Praktikum Statistik Industri (C)
2, 9	Praktikum Statistik Industri (A)

Jam Khusus dari dosen khusus (dosen LB)

Hari	Jam ke	Mata Kuliah
Senin	1 - 2	Teknik Peramalan (B)
	5 - 6	Statistik Industri II (A)
	7 - 8	APK (B)
		Statistik Industri I
Selasa		APK (A)
	5 - 6	MSDM (A)
	7 - 8	APK (B)
	11 - 12	Psikologi Industri (A)
Rabu	1 - 2	Statistik Industri II (B)
		Teknik Peramalan Lanjutan
	3 - 4	Kimia II
		Statistik Industri I

	5 - 6	Desain Eksperimen
	11 - 12	Etika Sosial
Kamis	7 - 8	SIM
		Ergonomi Industri
	11 - 12	Psikologi Industri (B)
Jumat	1 - 2	Teknik Peramalan Lanjutan
	5 - 6	Kalkulus I
	7 - 8	DSS
	9 - 10	Kalkulus I

LAMPIRAN C
DATA RUANG

LAMPIRAN C**Data Ruang**

Nama Ruang	Kapasitas
<u>Ruang kuliah :</u>	
B102	50
B105	50
B106	50
B107	50
B108	50
B111	50
B112	50
D101	50
D105	50
<u>Ruang Praktikum :</u>	
• C104 digunakan untuk praktikum pemodelan sistem, teknik peramalan, dan statistik industri.	20
• Lab Komputer digunakan untuk praktikum pemodelan sistem, teknik peramalan, dan statistik industri.	20
• C107 digunakan untuk praktikum fisika.	20
• FT-601 digunakan untuk praktikum TLP dan AKP.	20
• FT- 602 digunakan untuk praktikum TLP dan AKP.	20

LAMPIRAN D.
LISTING PROGRAM

unit Utama;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
Menus, ExtCtrls, ComCtrls, StdCtrls, Grids, DBGrids, Db, DBTables,
Buttons, Mask, DBCtrls;

type

```
TFormUtama = class(TForm)
  StatusBar: TStatusBar;
  Timer1: TTimer;
  MainMenu1: TMainMenu;
  File1: TMenuItem;
  Proses1: TMenuItem;
  N1: TMenuItem;
  Exit1: TMenuItem;
  Master1: TMenuItem;
  MataKuliah1: TMenuItem;
  Dosen1: TMenuItem;
  Ruang1: TMenuItem;
  DataJadual: TDataSource;
  Jadual: TTable;
  DataH_Dosen: TDataSource;
  H_Dosen: TTable;
  DataD_Dosen: TDataSource;
  D_Dosen: TTable;
  DataRuang: TDataSource;
  Ruang: TTable;
  Label4: TLabel;
  Label3: TLabel;
  Label2: TLabel;
  Label6: TLabel;
  PanelRuangan: TPanel;
  Label8: TLabel;
  Label9: TLabel;
  DBGridRuang: TDBGrid;
  OKRuang: TBitBtn;
  Ruangan: TEdit;
  Label11: TLabel;
  Kapasitas: TEdit;
  CloseRuang: TBitBtn;
  PanelMK: TPanel;
  Label12: TLabel;
  Label14: TLabel;
  Label15: TLabel;
  OKMK: TBitBtn;
  KodeMK: TEdit;
  NamaMK: TEdit;
  CloseMK: TBitBtn;
  Label16: TLabel;
  Label18: TLabel;
  Label19: TLabel;
  Label20: TLabel;
  Label21: TLabel;
  Label22: TLabel;
  SKS: TEdit;
```

Kotak: TEdit;
ClearMK: TBitBtn;
ClearRuang: TBitBtn;
Label23: TLabel;
Label24: TLabel;
Label17: TLabel;
JumMK: TLabel;
DeleteMK: TBitBtn;
Bevel1: TBevel;
Label13: TLabel;
JumRuang: TLabel;
DeleteRuang: TBitBtn;
Bevel3: TBevel;
PanelDosen: TPanel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label30: TLabel;
Label31: TLabel;
Label34: TLabel;
JumDosen: TLabel;
Bevel4: TBevel;
OKDs: TBitBtn;
KodeDs: TEdit;
NamaDs: TEdit;
CloseDs: TBitBtn;
ClearDs: TBitBtn;
DeleteDs: TBitBtn;
Label28: TLabel;
Label29: TLabel;
Smt: TEdit;
DataMatKul: TDataSource;
Label33: TLabel;
MatKul: TTable;
ViewDs: TButton;
QueryMasterDs: TQuery;
DataQueryMasterDs: TDataSource;
ListPilih2: TListBox;
ListSedia2: TListBox;
ListSediaTmp2: TListBox;
ListPilihTmp2: TListBox;
Label39: TLabel;
Label40: TLabel;
Pilih2: TBitBtn;
Batal2: TBitBtn;
Label11: TLabel;
Label15: TLabel;
DsnT: TRadioButton;
DsnK: TRadioButton;
UbahJadual1: TMenuItem;
PanelJadual: TPanel;
Label7: TLabel;
Label10: TLabel;
Label32: TLabel;
Label35: TLabel;
Label36: TLabel;
ListKodeMK: TComboBox;
Nama: TEdit;
GroupDsn1: TGroupBox;

```
Label37: TLabel;  
Label38: TLabel;  
Label41: TLabel;  
Label42: TLabel;  
KodeDsn1: TEdit;  
NamaDsn1: TEdit;  
Label43: TLabel;  
Label44: TLabel;  
GroupDsn2: TGroupBox;  
Label45: TLabel;  
Label46: TLabel;  
Label47: TLabel;  
Label48: TLabel;  
Label49: TLabel;  
Label50: TLabel;  
KodeDsn2: TEdit;  
NamaDsn2: TEdit;  
CloseJ: TBitBtn;  
ClearJ: TBitBtn;  
SaveJ: TBitBtn;  
GroupJadual: TGroupBox;  
Label51: TLabel;  
Label52: TLabel;  
Label54: TLabel;  
Label55: TLabel;  
HariKul: TComboBox;  
JamKul: TComboBox;  
Label53: TLabel;  
Label56: TLabel;  
JumDsn: TEdit;  
Label57: TLabel;  
Label58: TLabel;  
ListRuang: TComboBox;  
CariMK: TButton;  
StatusDsn1: TEdit;  
StatusDsn2: TEdit;  
CariDsn1: TButton;  
CariDsn2: TButton;  
PanelCariMK: TPanel;  
DBGridCariMK: TDBGrid;  
Label59: TLabel;  
PilihMK: TBitBtn;  
BatalMK: TBitBtn;  
QueryJadual: TQuery;  
Label61: TLabel;  
Label60: TLabel;  
KuliahKe: TComboBox;  
Label62: TLabel;  
Label63: TLabel;  
Satu: TRadioButton;  
Dua: TRadioButton;  
QueryKotak: TQuery;  
D_Ruang: TTable;  
Label64: TLabel;  
QueryRuang: TQuery;  
PanelCariDsn: TPanel;  
Label65: TLabel;  
DBGridCariDsn: TDBGrid;  
PilihDsn: TBitBtn;
```

```

BatalDsn: TBitBtn;
DosenKe: TLabel;
ViewMK: TButton;
CariKodeMK: TButton;
ListRSedia: TListBox;
ListRPilih: TListBox;
Label66: TLabel;
Label67: TLabel;
PilihR: TBitBtn;
BatalR: TBitBtn;
Caller: TLabel;
CariDsn: TButton;
Caller2: TLabel;
QueryDosen: TQuery;
DataQueryDosen: TDataSource;
DBGridCariDsn2: TDBGrid;
QueryHitung: TQuery;
procedure Timer1Timer(Sender: TObject);
procedure OKRuangClick(Sender: TObject);
procedure CloseRuangClick(Sender: TObject);
procedure Ruang1Click(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure ClearRuangClick(Sender: TObject);
procedure MataKuliah1Click(Sender: TObject);
procedure CloseMKClick(Sender: TObject);
procedure ClearMKClick(Sender: TObject);
procedure OKMKClick(Sender: TObject);
procedure DeleteMKClick(Sender: TObject);
procedure Dosen1Click(Sender: TObject);
procedure OKDsClick(Sender: TObject);
procedure CloseDsClick(Sender: TObject);
procedure ClearDsClick(Sender: TObject);
procedure DeleteDsClick(Sender: TObject);
procedure Proses1Click(Sender: TObject);
procedure ViewDsClick(Sender: TObject);
procedure KodeDsChange(Sender: TObject);
procedure Pilih2Click(Sender: TObject);
procedure Batal2Click(Sender: TObject);
procedure KodeMKChange(Sender: TObject);
procedure ClearJClick(Sender: TObject);
procedure UbahJadual1Click(Sender: TObject);
procedure CloseJClick(Sender: TObject);
procedure PilihMKClick(Sender: TObject);
procedure CariMKClick(Sender: TObject);
procedure BatalMKClick(Sender: TObject);
procedure DBGridCariMKDblClick(Sender: TObject);
procedure ListKodeMKChange(Sender: TObject);
procedure KuliahKeChange(Sender: TObject);
procedure PilihDsnClick(Sender: TObject);
procedure CariDsn1Click(Sender: TObject);
procedure CariDsn2Click(Sender: TObject);
procedure BatalDsnClick(Sender: TObject);
procedure DBGridCariDsnDblClick(Sender: TObject);
procedure SaveJClick(Sender: TObject);
procedure ViewMKClick(Sender: TObject);
procedure CariKodeMKClick(Sender: TObject);
procedure PilihRClick(Sender: TObject);
procedure BatalRClick(Sender: TObject);
procedure DeleteRuangClick(Sender: TObject);

```

```

    procedure DBGridRuangCellClick(Column: TColumn);
    procedure RuanganChange(Sender: TObject);
    procedure CariDsnClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

type
    TMatKul = record
        kodemk : string[11];
        namamk : string[22];
        kodeds : string[5];
        ruang : string[5];
        hari : byte;
        jam : byte;
        jumdsn : byte;
        fixed : boolean;
    end;

const
    max = 24;
    hari : array [1..7] of string = ('Minggu', 'Senin', 'Selasa',
    'Rabu', 'Kamis', 'Jumat', 'Sabtu');
    bulan : array [1..12] of string = ('Januari', 'Februari', 'Maret',
    'April', 'Mei', 'Juni', 'Juli', 'Agustus', 'September',
    'Oktober', 'November', 'Desember');

var
    FormUtama: TFormUtama;
    kul : array of TMatKul;
    t_hari, t_jam : byte;

implementation

uses Proses, Cetak;
{$R *.DFM}

procedure TFormUtama.Timer1Timer(Sender: TObject);
var
    dd, mm, yy : word;
begin
    DecodeDate(Now, yy, mm, dd);
    StatusBar.Panels.Items[0].Text := hari[DayOfWeek(Now)] + ', ' +
    IntToStr(dd) + ' ' + bulan[mm] + ' ' + IntToStr(yy);
    StatusBar.Panels.Items[1].Text := TimeToStr(Now);
end;

//*****
//***          Master Mata Kuliah          ***
//*****
procedure TFormUtama.MataKuliah1Click(Sender: TObject);
begin
    Master1.Enabled := false;
    PanelMK.show;
    JumMK.Caption := IntToStr(MatKul.RecordCount);
    KodeMK.SetFocus;

```



```

end;

procedure TFormUtama.OKMKClick(Sender: TObject);
var
    valid,nemu,ganti : boolean;
    i                 : byte;
begin
    if (length(KodeMK.Text)<11) then
        begin
            messagedlg('Kode mata kuliah harus 11
digit!',mtinformation,[mbOK],0);
            KodeMK.SelectAll;
            KodeMK.SetFocus;
        end
    else if (ListRPilih.Items.Count=0) then
        messagedlg('Anda harus memilih ruangan yang
diperlukan!',mtinformation,[mbOK],0)
    else
        begin
            valid := true;
            if not (SKS.Text[1] in ['0'..'9']) or not (Kotak.Text[1] in
['0'..'9']) or
                not (Smt.Text[1] in ['0'..'9']) then valid := false;
            if valid then
                begin
                    if MatKul.FindKey([uppercase(KodeMK.Text)]) then

                        begin
                            MatKul.Locate('KodeMK',
vararrayof([uppercase(KodeMK.Text)]), []);
                            MatKul.edit;
                        end
                    else MatKul.append;
                    MatKul.FieldValues['KodeMK'] := uppercase(KodeMK.Text);
                    MatKul.FieldValues['NamaMK'] := NamaMK.Text;
                    MatKul.FieldValues['SKS']   := StrToInt(SKS.Text);
                    MatKul.FieldValues['Kotak']  := StrToInt(Kotak.Text);
                    MatKul.FieldValues['Smt']    := Smt.Text;
                    if (Satu.Checked) then
                        MatKul.FieldValues['JumDosen'] := 1
                    else
                        MatKul.FieldValues['JumDosen'] := 2;
                    MatKul.Post;
                    MatKul.Close;
                    MatKul.Open;
                    MatKul.Refresh;
                    JumMK.Caption := IntToStr(MatKul.RecordCount);
                    QueryKotak.ParamByName('tkode').asstring :=
uppercase(KodeMK.Text);
                    QueryKotak.Open;
                    for i := 1 to (StrToInt(Kotak.Text[1])-
QueryKotak.RecordCount) do
                        begin
                            Jadual.Append;
                            Jadual.FieldValues['kodemk'] := uppercase(KodeMK.Text);
                            Jadual.Post;
                        end;
                    QueryKotak.Close;
                    Jadual.Close;
                end
            end
        end
    end
end;

```

```

Jadual.Open;
Jadual.Refresh;
D_Ruang.First;
while not D_Ruang.eof do
    begin
        if
D_Ruang.fieldbyname('kodemk').asString=uppercase(KodeMK.Text) then
            D_Ruang.delete
        else D_Ruang.next;
        end;
        {cek apakah ruang pada tabel jadual perlu di-update}
        Jadual.First; nemu := false;
        while not Jadual.Eof and not nemu do
            begin
                if (Jadual.FieldValues['kodemk']=uppercase(KodeMK.Text))
then
                    nemu := true
                else Jadual.Next;
                end;
                ganti := true;
                for i := 0 to ListRPilih.Items.Count-1 do
                    begin
                        if nemu and
(Jadual.FieldValues['kodemk']=ListRPilih.Items[i]) then
                            ganti := false;
                            D_Ruang.Append;
                            D_Ruang.FieldValues['kodemk'] := uppercase(KodeMK.Text);
                            D_Ruang.FieldValues['koder'] := ListRPilih.Items[i];
                            D_Ruang.Post;
                        end;
                        if nemu and ganti then
                            begin
                                Jadual.First;
                                while not Jadual.Eof do
                                    begin
                                        if
(Jadual.FieldValues['kodemk']=uppercase(KodeMK.Text)) then
                                            begin
                                                Jadual.Edit;
                                                Jadual.FieldValues['ruang'] :=
ListRPilih.Items[0];
                                                Jadual.Post;
                                            end;
                                                Jadual.Next;
                                            end;
                                        end;
                                    end;
                                D_Ruang.Refresh;
                                D_Ruang.Close;
                                D_Ruang.Open;
                                Jadual.Close;
                                Jadual.Open;
                                JumMK.Caption := IntToStr(MatKul.RecordCount);
                                ClearMK.Click;
                                OKMK.Caption := 'Save';
                            end
                        else
                            begin
                                messagedlg('SKS dan Kotak harus berisi
angka!', mtinformation, [mbOK], 0);

```

```

        SKS.SetFocus;
    end;
end;
end;

* procedure TFormUtama.DeleteMKClick(Sender: TObject);
begin
    if MatKul.findkey([uppercase(KodeMK.text)]) then
        begin
            if MessageDlg('Data mau dihapus?', mtConfirmation, [mbYes,
mbNo], 0) = mrYes then
                begin
                    D_Ruang.First;
                    while not D_Ruang.Eof do
                        begin
                            if (D_Ruang.FieldValues['kodeMK']=uppercase(KodeMK.text))
then
                                D_Ruang.Delete
                            else D_Ruang.Next;
                            end;
                            Jadual.First;
                            while not Jadual.Eof do
                                begin
                                    if (Jadual.FieldValues['kodeMK']=uppercase(KodeMK.text))
then
                                        Jadual.Delete
                                    else Jadual.Next;
                                    end;
                                D_Ruang.Close;
                                D_Ruang.Open;
                                Jadual.Close;
                                Jadual.Open;
                                MatKul.Delete;
                                ClearMK.Click;
                                MessageDlg('Data telah dihapus.', mtInformation, [mbOK], 0);
                                end;
                            end
                        else
                            begin
                                messagedlg('Data tidak ditemukan!', mtinformation, [mbOK], 0);
                                KodeMK.SelectAll;
                                end;
                            end;
end;

* procedure TFormUtama.ClearMKClick(Sender: TObject);
begin
    KodeMK.Text := '';
    NamaMK.Text := '';
    SKS.Text := '';
    Kotak.Text := '';
    Smt.Text := '';
    KodeMK.SetFocus;
    OKMK.Caption := 'Save';
    DeleteMK.Enabled := false;
end;

* procedure TFormUtama.CloseMKClick(Sender: TObject);
begin
    ClearMK.Click;

```

```

PanelMK.Visible := false;
Master1.Enabled := true;
end;

```

```

//*****
//***                               Master Ruang                               ***
//*****

```

```

procedure TFormUtama.Ruang1Click(Sender: TObject);
begin
    Master1.Enabled := false;
    PanelRuangan.show;
    JumRuang.Caption := IntToStr(Ruang.RecordCount);
end;

```

```

procedure TFormUtama.OKRuangClick(Sender: TObject);

```

```

var
    i      : byte;
    valid  : boolean;
begin
    if (length(Ruangan.Text)<4) then
        begin
            messagedlg('Ruangan harus 4 digit!',mtinformation,[mbOK],0);
            Ruangan.SelectAll;
            Ruangan.SetFocus;
        end
    else
        begin
            valid := true; i := 0;
            repeat
                inc(i);
                if not (Kapasitas.Text[i] in ['0'..'9']) then valid := false;
            until not valid or (i=length(Kapasitas.Text));
            if valid then
                begin
                    if Ruang.findkey([uppercase(Ruangan.text)]) then
                        Ruang.Edit
                    else Ruang.Append;
                    Ruang.FieldName('KodeR').AsString :=
uppercase(Ruangan.Text);
                    Ruang.FieldName('Kapasitas').AsInteger :=
StrToInt(Kapasitas.Text);
                    Ruang.Post;
                    Ruang.Refresh;
                    Ruang.Close;
                    Ruang.Open;
                    JumRuang.Caption := IntToStr(Ruang.RecordCount);
                    ClearRuang.Click;
                end
            else
                begin
                    messagedlg('Kapasitas ruangan harus berisi
angka!',mtinformation,[mbOK],0);
                    Kapasitas.SelectAll;
                    Kapasitas.SetFocus;
                end;
            end;
        end;
    end;
end;

```

```

procedure TFormUtama.CloseRuangClick(Sender: TObject);
begin
    ClearRuang.Click;
    PanelRuangan.hide;
    Master1.Enabled := true;
end;

procedure TFormUtama.ClearRuangClick(Sender: TObject);
begin
    Ruangan.Text := '';
    Kapasitas.Text := '0';
    Ruangan.SetFocus;
    DeleteRuang.Enabled := false;
end;

procedure TFormUtama.DeleteRuangClick(Sender: TObject);
var
    nemu : boolean;
begin
    if Ruang.FindKey([uppercase(Ruangan.Text)]) then
        begin
            if MessageDlg('Data mau dihapus?', mtConfirmation, [mbYes,
mbNo], 0) = mrYes
                then begin
                    Jadual.First; nemu := false;
                    while not Jadual.eof and not nemu do
                        begin
                            if (Jadual.FieldValues['ruang']=uppercase(Ruangan.Text))
                                then
                                    nemu := true;
                                    Jadual.next;
                                end;
                            if nemu then
                                begin
                                    messagedlg('Ruang '+uppercase(Ruangan.Text)+' masih
dipakai!', mtInformation, [mbOK], 0);
                                    end
                                else //hapus ruang pada tabel ruang & d_ruang yg sesuai
                                    begin
                                        Ruang.Delete;
                                        Ruang.Refresh;
                                        Ruang.Close;
                                        Ruang.Open;
                                        ClearRuang.Click;
                                        D_Ruang.First;
                                        while not D_Ruang.eof do
                                            begin
                                                if
(D_Ruang.FieldValues['koder']=uppercase(Ruangan.Text)) then
                                                    D_Ruang.Delete
                                                else D_Ruang.Next;
                                            end;
                                                MessageDlg('Data telah dihapus.',
mtInformation, [mbOk], 0);
                                                D_Ruang.Close;
                                                D_Ruang.Open;
                                            end;
                                        end;
                                    end;
                            end;
                    end;
                end;
        end;
end;

```

```

    end;
end;

procedure TFormUtama.DBGridRuangCellClick(Column: TColumn);
begin
    Ruangan.Text := DBGridRuang.Fields[0].asString;
end;

procedure TFormUtama.RuanganChange(Sender: TObject);
begin
    Kapasitas.Text := '0';
    if (Ruang.Findkey([uppercase(Ruangan.Text)])) then
    begin
        Kapasitas.Text := Ruang.FieldName('kapasitas').asString;
        OKRuang.Caption := 'Update';
        DeleteRuang.Enabled := true;
    end
    else
    begin
        OKRuang.Caption := 'Save';
        DeleteRuang.Enabled := false;
    end;
end;

//*****
//***          Master Dosen          ***
//*****
procedure TFormUtama.Dosen1Click(Sender: TObject);
begin
    Master1.Enabled := false;
    PanelDosen.show;
    JumDosen.Caption := IntToStr(H_Dosen.RecordCount);
    KodeDs.SetFocus;
end;

procedure TFormUtama.OKDsClick(Sender: TObject);
var
    i : byte;
begin
    if (length(KodeDs.Text)<5) then
    begin
        messagedlg('Kode Dosen harus 5 digit!',mtinformation,[mbOK],0);
        KodeDs.SelectAll;
    end
    else
    begin
        if H_Dosen.FindKey([uppercase(KodeDs.Text)]) then H_Dosen.edit
        else H_Dosen.append;
        H_Dosen.FieldValues['KodeDs'] := uppercase(KodeDs.Text);
        H_Dosen.FieldValues['namads'] := NamaDs.Text;
        if (DsnT.Checked=true) then
            H_Dosen.FieldName('status').asboolean := false
        else if (DsnK.Checked=true) then
            H_Dosen.FieldName('status').asboolean := true;
        H_Dosen.post;
        H_Dosen.close;
        H_Dosen.open;
        D_Dosen.First;
    end;
end;

```

```

while not D_Dosen.eof do
begin
    if
D_Dosen.fieldbyname('kodeds').asString=uppercase(KodeDs.Text) then
        D_Dosen.delete
    else D_Dosen.next;
end;
for i := 0 to ListPilih2.Items.Count-1 do
begin
    D_Dosen.Append;
    D_Dosen.FieldValues['kodeds'] := uppercase(KodeDs.Text);
    D_Dosen.FieldValues['kodemk'] := ListPilihTmp2.Items[i];
    D_Dosen.Post;
end;
D_Dosen.close;
D_Dosen.open;
JumDosen.Caption := IntToStr(H_Dosen.RecordCount);
ClearDs.Click;
OKDs.Caption := 'Save';
end;
end;

procedure TFormUtama.DeleteDsClick(Sender: TObject);
var
    nemu : boolean;
begin
    if H_Dosen.findkey([uppercase(KodeDs.text)]) then
begin
        if MessageDlg('Data mau dihapus?',mtConfirmation,[mbYes,
mbNo],0)=mrYes then
            begin
                Jadual.First; nemu := false;
                while not Jadual.eof and not nemu do
                    begin
                        if (Jadual.FieldValues['kodeds']=uppercase(KodeDs.Text))
then
                            nemu := true;
                            Jadual.next;
                        end;
                    if nemu then
                        begin
                            messagedlg('Dosen '+uppercase(KodeDs.Text)+' masih
aktif!',mtinformation,[mbOK],0);
                        end
                    else //hapus ruang pada tabel h_dosen & d_dosen yg sesuai
                        begin
                            H_Dosen.Delete;
                            H_Dosen.close;
                            H_Dosen.open;
                            D_Dosen.First;
                            while not D_Dosen.eof do
                                begin
                                    if
D_Dosen.fieldbyname('kodeds').asString=uppercase(KodeDs.Text) then
                                        D_Dosen.delete
                                    else D_Dosen.next;
                                end;
                            D_Dosen.close;
                            D_Dosen.open;

```

```

        JumDosen.Caption := IntToStr(H_Dosen.RecordCount);
        MessageDlg('Data telah dihapus.', mtInformation, [mbOk], 0);
        ClearDs.Click;
        OKDs.Caption := 'Save';
    end;
end;
end
else
begin
    messagedlg('Data tidak ditemukan!', mtInformation, [mbOk], 0);
    KodeDs.SelectAll;
end;
end;

procedure TFormUtama.CloseDsClick(Sender: TObject);
begin
    PanelDosen.Visible := false;
    KodeDs.Text := '';
    NamaDs.Text := '';
    DsnT.Checked := true;
    DeleteDs.Enabled := false;
    Master1.Enabled := true;
end;

procedure TFormUtama.ClearDsClick(Sender: TObject);
begin
    KodeDs.Text := '';
    NamaDs.Text := '';
    DsnT.Checked := true;
    KodeDs.SetFocus;
    DeleteDs.Enabled := false;
end;

//Keluar Program
procedure TFormUtama.Exit1Click(Sender: TObject);
begin
    close;
end;

procedure TFormUtama.Proses1Click(Sender: TObject);
begin
    FormProses.visible := true;
    FormUtama.Enabled := false;
end;

procedure TFormUtama.ViewDsClick(Sender: TObject);
begin
    formcetak.queryds.open;
    formcetak.queryrep.open;
    formcetak.LaporanDsn.preview;
end;

procedure TFormUtama.KodeDsChange(Sender: TObject);
var
    i : byte;
    nemu : boolean;
begin
    ListSedia2.Clear; ListSediaTmp2.Clear;
    ListPilih2.Clear; ListPilihTmp2.Clear;

```



```

DeleteDs.Enabled := false;
NamaDs.Text := '';
if (length(KodeDs.Text)=5) then
begin
    MatKul.First;
    while not MatKul.eof do
        begin
listSediaTmp2.Items.add(MatKul.fieldbyname('kodemk').asString);
            listSedia2.Items.add(MatKul.fieldbyname('namamk').asString+'
('+MatKul.FieldName('sks').asString+' SKS)');
            MatKul.next;
        end;
    QueryMasterDs.close;
    QueryMasterDs.parambyname('kode').asString :=
uppercase(KodeDs.Text);
    QueryMasterDs.open;
    QueryMasterDs.First;
    if H_Dosen.FindKey([uppercase(KodeDs.Text)]) then
        begin
            OKDs.Caption := 'Update';
            NamaDs.Text := H_Dosen.FieldValues['namads'];
            if (H_Dosen.FieldValues['status']=true) then
                DsnK.Checked := true
            else
                DsnT.Checked := true;
            DeleteDs.Enabled := true;
        end
    else OKDs.Caption := 'Save';
    while not(QueryMasterDs.Eof) do
        begin

ListPilih2.Items.Add(QueryMasterDs.FieldName('namamk').asString);

ListPilihTmp2.Items.Add(QueryMasterDs.FieldName('kodemk').asString);
        i := 0; nemu := false;
        if ListSediaTmp2.Items.Count>0 then
            repeat
                if
(ListSediaTmp2.Items[i]=QueryMasterDs.FieldName('kodemk').asString)
then
                    begin
                        nemu := true;
                        ListSedia2.Items.Delete(i);
                        ListSediaTmp2.Items.Delete(i);
                    end;
                inc(i);
            until (i=ListSedia2.Items.Count) or nemu;
            QueryMasterDs.Next;
        end;
    end;
end;

procedure TFormUtama.Pilih2Click(Sender: TObject);
var
    ketemu : boolean;
begin
    if (ListSedia2.ItemIndex > -1) then
        begin

```

```

MatKul.First; ketemu := false;
while not ketemu do
begin
    if
(MatKul.FieldName('kodemk').asString=ListSediaTmp2.Items[ListSedia2.
ItemIndex])
        then ketemu := true;
        MatKul.Next;
    end;
    DeleteDs.enabled := false;
    ListPilih2.Items.Add(ListSedia2.Items[ListSedia2.ItemIndex]);

ListPilihTmp2.Items.Add(ListSediaTmp2.Items[ListSedia2.ItemIndex]);
    ListSediaTmp2.Items.Delete(ListSedia2.ItemIndex);
    ListSedia2.Items.Delete(ListSedia2.ItemIndex);
end
else showmessage('Anda belum memilih mata kuliah pada kotak sebelah
kiri');
end;

procedure TFormUtama.Batal2Click(Sender: TObject);
var
    ketemu : boolean;
begin
    if (ListPilih2.ItemIndex > -1) then
    begin
        DeleteDs.enabled := false;
        MatKul.First; ketemu := false;
        while not ketemu do
        begin
            if
(MatKul.FieldName('kodemk').asString=ListPilihTmp2.Items[ListPilih2.
ItemIndex])
                then ketemu := true;
                MatKul.Next;
            end;
            ListSedia2.Items.Add(ListPilih2.Items[ListPilih2.ItemIndex]);

ListSediaTmp2.Items.Add(ListPilihTmp2.Items[ListPilih2.ItemIndex]);
            ListPilihTmp2.Items.Delete(ListPilih2.ItemIndex);
            ListPilih2.Items.Delete(ListPilih2.ItemIndex);
        end
        else showmessage('Anda belum memilih mata kuliah pada kotak sebelah
kanan');
    end;

procedure TFormUtama.KodeMKChange(Sender: TObject);
var
    i      : byte;
    nemu   : boolean;
begin
    ListRSedia.Clear; ListRPilih.Clear;
    DeleteMK.Enabled := false;
    NamaMK.Text := '';
    SKS.Text    := '';
    Kotak.Text  := '';
    Smt.Text    := '';
    Satu.Checked := true;
    if (length(KodeMK.Text)=11) then

```

```

begin
  Ruang.First;
  while not Ruang.eof do
    begin
      listRSedia.Items.add(Ruang.fieldbyname('koder').asstring);
      Ruang.next;
    end;
  if MatKul.FindKey([uppercase(KodeMK.Text)]) then
    begin
      OKMK.Caption := 'Update';
      NamaMK.Text := MatKul.FieldValues['namamk'];
      SKS.Text := MatKul.FieldValues['sks'];
      Kotak.Text := MatKul.FieldValues['kotak'];
      Smt.Text := MatKul.FieldValues['smt'];
      if (MatKul.FieldValues['jumdosen']=1) then
        Satu.Checked := true
      else
        Dua.Checked := true;
        DeleteMK.Enabled := true;
      end
    else OKMK.Caption := 'Save';
    QueryRuang.Close;
    QueryRuang.ParamByName('tkode').asstring :=
uppercase(KodeMK.Text);
    QueryRuang.Open;
    QueryRuang.First;
    while not(QueryRuang.Eof) do
      begin
        ListRPilih.Items.Add(QueryRuang.FieldValues['koder']);
        i := 0; nemu := false;
        if ListRSedia.Items.Count>0 then
          repeat
            if (ListRSedia.Items[i]=QueryRuang.FieldValues['koder']) then
              begin
                nemu := true;
                ListRSedia.Items.Delete(i);
              end;
            inc(i);
          until (i=ListRSedia.Items.Count) or nemu;
          QueryRuang.Next;
        end;
      end;
    end;
  end;

procedure TFormUtama.ClearJClick(Sender: TObject);
begin
  ListKodeMK.Text := '';
  Nama.Text := '';
  KuliahKe.Items.Clear;
  ListRuang.Itemindex := -1;
  JumDsn.Text := '';
  KodeDsn1.Text := '';
  NamaDsn1.Text := '';
  StatusDsn1.Text := '';
  KodeDsn2.Text := '';
  NamaDsn2.Text := '';
  StatusDsn2.Text := '';
  HariKul.Itemindex := -1;
  JamKul.Itemindex := -1;

```

```

end;

procedure TFormUtama.UbahJadual1Click(Sender: TObject);
begin
    PanelJadual.Show;
    MatKul.First;
    while not MatKul.Eof do
        begin
            ListKodeMK.Items.Add(MatKul.FieldValues['kode_mk']);
            MatKul.Next;
        end;
    end;

procedure TFormUtama.CloseJClick(Sender: TObject);
begin
    ListKodeMK.Items.Clear;
    ListRuang.Items.Clear;
    ClearJ.Click;
    PanelJadual.Hide;
end;

procedure TFormUtama.PilihMKClick(Sender: TObject);
begin
    if (Caller.Caption='PanelJadual') then
        begin
            PanelCariMK.Hide;
            PanelJadual.Enabled := true;
            ClearJ.Click;
            ListKodeMK.ItemIndex := MatKul.RecNo-1;
            ListKodeMK.OnChange(sender);
        end
    else if (Caller.Caption='PanelMK') then
        begin
            PanelCariMK.Hide;
            PanelMK.Enabled := true;
            ClearMK.Click;
            KodeMK.Text := DBGridCariMK.Fields[0].AsString;
            KodeMK.OnChange(sender);
        end;
    end;

procedure TFormUtama.CariMKClick(Sender: TObject);
begin
    MatKul.First;
    PanelCariMK.Show;
    PanelJadual.Enabled := false;
    Caller.Caption := 'PanelJadual';
end;

procedure TFormUtama.BatalMKClick(Sender: TObject);
begin
    PanelCariMK.Hide;
    if (Caller.Caption='PanelJadual') then PanelJadual.Enabled := true
    else PanelMK.Enabled := true;
end;

procedure TFormUtama.DBGridCariMKDblClick(Sender: TObject);
begin
    PilihMK.Click;

```

```

end;

procedure TFormUtama.ListKodeMKChange(Sender: TObject);
var
  tmp : string;
  i : byte;
begin
  tmp := ListKodeMK.Text;
  ClearJ.Click;
  ListKodeMK.Text := tmp;
  if MatKul.FindKey([ListKodeMK.Text]) then
    begin
      QueryDosen.ParamByName('tkode').asString :=
ListKodeMK.Items[ListKodeMK.Itemindex];
      DBGridCariDsn2.Columns[0].FieldName := 'KodeDs';
      DBGridCariDsn2.Columns[1].FieldName := 'NamaDs';
      QueryJadual.ParamByName('tkode').AsString := ListKodeMK.Text;
      QueryJadual.Open;
      setlength(Kul,QueryJadual.RecordCount+1);
      for i := 1 to QueryJadual.RecordCount do
        KuliAhKe.Items.Add(inttostr(i));
      i := 0;
      while not QueryJadual.Eof do
        begin
          inc(i);
          Kul[i].kodeMK := ListKodeMK.Text;
          Kul[i].namamk := Nama.Text;
          Kul[i].ruang := QueryJadual.FieldName('ruang').asString;
          Kul[i].hari := QueryJadual.FieldName('hari').asinteger;
          Kul[i].jam := QueryJadual.FieldName('jam').asinteger;
          Kul[i].fixed := QueryJadual.FieldName('fixed').asboolean;
          Kul[i].kodeds := QueryJadual.FieldName('kodeds').asString;
          Kul[i].jumdsn := MatKul.FieldName('jumdosen').asinteger;
          QueryJadual.Next;
        end;
      QueryJadual.Close;
      KuliAhKe.Text := KuliAhKe.Items[0];
      KuliAhKe.Enabled := true;
      Nama.Text := MatKul.FieldName('namamk').asString;
    end
  else //ndak nemu
    begin
      ListKodeMK.SelectAll;
      KuliAhKe.Enabled := false;
    end;
end;

```

```

procedure TFormUtama.KuliAhKeChange(Sender: TObject);
var
  i : byte;
  nemu : boolean;
  tmp,dsn1,dsn2 : string;
begin
  if (KuliAhKe.Text<>'') then
    begin
      ListRuang.Items.Clear;
      QueryRuang.ParamByName('tkode').asString := ListKodeMK.Text;
      QueryRuang.Open;
      QueryRuang.First;

```

```

while not QueryRuang.Eof do

    begin
        ListRuang.Items.Add(QueryRuang.FieldValues['koder']);
        QueryRuang.Next;
    end;
QueryRuang.Close;
i := 0; nemu := false;
repeat
    if (ListRuang.Items[i]=Kul[strtoint(KuliahKe.Text)].ruang) then
        nemu := true
    else inc(i);
until (i=ListRuang.Items.Count) or nemu;
ListRuang.Itemindex := i;
JumDsn.Text      := inttostr(Kul[strtoint(KuliahKe.Text)].jumdsn);
HariKul.Itemindex := Kul[strtoint(KuliahKe.Text)].hari-1;
JamKul.Itemindex  := Kul[strtoint(KuliahKe.Text)].jam-1;
if (HariKul.Itemindex>-1) and (JamKul.Itemindex>-1) then
    begin
        t_hari := HariKul.Itemindex+1;
        t_jam  := JamKul.Itemindex+1;
    end;
tmp := Kul[strtoint(KuliahKe.Text)].kodeds;
KodeDsn1.Text  := '';
NamaDsn1.Text  := '';
StatusDsn1.Text := '';
KodeDsn2.Text  := '';
NamaDsn2.Text  := '';
StatusDsn2.Text := '';
if (JumDsn.Text='1') then //1 dosen
    begin
        GroupDsn2.Enabled := false;
        if (tmp<>'') then
            begin
                dsn1 := tmp;
                KodeDsn1.Text := dsn1;
                H_Dosen.FindKey([dsn1]);
                NamaDsn1.Text := H_Dosen.FieldValues['namads'];
                if (H_Dosen.FieldValues['status']=false) then
                    StatusDsn1.Text := 'Dosen Tetap'
                else
                    StatusDsn1.Text := 'Dosen Khusus';
            end;
        end;
    end
else //2 dosen
    begin
        GroupDsn2.Enabled := true;
        if (tmp<>'') then
            begin
                dsn1 := 'DS0' + copy(tmp,2,2);
                dsn2 := 'DS0' + copy(tmp,4,2);
                KodeDsn1.Text := dsn1;
                H_Dosen.FindKey([dsn1]);
                NamaDsn1.Text := H_Dosen.FieldValues['namads'];
                if (H_Dosen.FieldValues['status']=false) then
                    StatusDsn1.Text := 'Dosen Tetap'
                else
                    StatusDsn1.Text := 'Dosen Khusus';
                KodeDsn2.Text := dsn2;
            end;
        end;
    end;
end;

```

```

        H_Dosen.FindKey([dsn2]);
        NamaDsn2.Text := H_Dosen.FieldValues['namads'];
        if (H_Dosen.FieldValues['status']=false) then
            StatusDsn2.Text := 'Dosen Tetap'
        else
            StatusDsn2.Text := 'Dosen Khusus';
        end;
    end;
end;
end;
end;

procedure TFormUtama.PilihDsnClick(Sender: TObject);
begin
    if (Caller2.Caption='PanelDosen') then
        begin
            KodeDs.Text := DBGridCariDsn.Fields[0].AsString;
            DBGridCariDsn.hide;
            PanelDosen.Enabled := true;
        end
    else if (Caller2.Caption='PanelJadual') then
        begin
            if (DosenKe.Caption='1') then
                begin
                    KodeDsn1.Text := DBGridCariDsn2.Fields[0].AsString;
                    NamaDsn1.Text := DBGridCariDsn2.Fields[1].AsString;
                    H_Dosen.FindKey([KodeDsn1.Text]);
                    if (H_Dosen.Fieldbyname('status').asboolean=true) then
                        StatusDsn1.Text := 'Dosen Tetap'
                    else
                        StatusDsn1.Text := 'Dosen Khusus';
                    end
                end
            else
                begin
                    KodeDsn2.Text := DBGridCariDsn2.Fields[0].AsString;
                    NamaDsn2.Text := DBGridCariDsn2.Fields[1].AsString;
                    H_Dosen.FindKey([KodeDsn2.Text]);
                    if (H_Dosen.Fieldbyname('status').asboolean=true) then
                        StatusDsn2.Text := 'Dosen Tetap'
                    else
                        StatusDsn2.Text := 'Dosen Khusus';
                    end;
                end;
            DBGridCariDsn2.hide;
            QueryDosen.Close;
        end;
        PanelCariDsn.Hide;
        PanelJadual.Enabled := true;
    end;

procedure TFormUtama.CariDsn1Click(Sender: TObject);
begin
    QueryDosen.Open;
    QueryDosen.First;
    PanelCariDsn.Show;
    DBGridCariDsn2.show;
    DosenKe.Caption := '1';
    PanelJadual.Enabled := false;
    Caller2.Caption := 'PanelJadual';
end;

```

```

procedure TFormUtama.CariDsn2Click(Sender: TObject);
begin
    QueryDosen.Open;
    QueryDosen.First;
    PanelCariDsn.Show;
    DBGridCariDsn2.show;
    DosenKe.Caption := '2';
    PanelJadual.Enabled := false;
    Caller2.Caption := 'PanelJadual';
end;

```

```

procedure TFormUtama.BatalDsnClick(Sender: TObject);
begin
    if (Caller2.Caption='PanelJadual') then
    begin
        PanelJadual.Enabled := true;
        QueryDosen.Close;
        DBGridCariDsn2.hide;
    end
    else if (Caller2.Caption='PanelDosen') then
    begin
        PanelDosen.Enabled := true;
        DBGridCariDsn.hide;
    end;
    PanelCariDsn.Hide;
end;

```

```

procedure TFormUtama.DBGridCariDsnDblClick(Sender: TObject);
begin
    PilihDsn.Click;
end;

```

```

procedure TFormUtama.SaveJClick(Sender: TObject);
var
    valid : boolean;
    i,n    : byte;
begin
    if (length(ListKodeMK.Text)<11) then
    begin
        messagedlg('Kode MK harus 11 digit!',mtinformation,[mbOK],0);
        ListKodeMK.SelectAll;
    end
    else if (HariKul.Text='') then
    begin
        messagedlg('Hari kuliah harus terisi!',mtinformation,[mbOK],0);
        HariKul.SelectAll;
    end
    else if (JamKul.Text='') then
    begin
        messagedlg('Jam kuliah harus terisi!',mtinformation,[mbOK],0);
        JamKul.SelectAll;
    end
    else begin
        valid := true;
        if ((JumDsn.Text='1') and (KodeDsn1.Text='')) or
            ((JumDsn.Text='2') and ((KodeDsn1.Text='') or
            (KodeDsn2.Text='')))
        then begin
            valid := false;

```



```

        messagedlg('Data dosen harus terisi!',mtinformation,[mbOK],0);
    end
    else if valid then
        begin
            QueryHitung.ParamByName('thari').AsInteger :=
HariKul.ItemIndex+1;
            QueryHitung.ParamByName('tjam').AsInteger :=
JamKul.ItemIndex+1;
            QueryHitung.Open;
            if (t_hari=HariKul.ItemIndex+1) and (t_jam=JamKul.ItemIndex+1)
then
                n := 1
            else n := 0;
            if (QueryHitung.RecordCount=6+n) then
                begin
                    valid := false;
                    messagedlg('Jadual sudah penuh! Pilihlah jadual
lainnya.',mtinformation,[mbOK],0);
                end;
                QueryHitung.Close;
            end;
            if valid then
                begin
                    Jadual.First; i := 0;
                    while not (Jadual.Eof) and (i<KuliahKe.Items.Count) do
                        begin
                            if (Jadual.FieldValues['kodemk']=ListKodeMK.Text) then
                                begin
                                    inc(i);
                                    Jadual.Edit;
                                    Jadual.FieldValues['ruang'] := ListRuang.Text;
                                    if (i=strtoint(KuliahKe.Text)) then
                                        begin
                                            Jadual.FieldValues['hari'] := HariKul.ItemIndex+1;
                                            Jadual.FieldValues['jam'] := JamKul.ItemIndex+1;
                                            if (JumDsn.Text='1') then
                                                Jadual.FieldValues['kodeds'] := KodeDsn1.Text
                                            else
                                                Jadual.FieldValues['kodeds'] :=
'D'+copy(KodeDsn1.Text,4,2)+copy(KodeDsn2.Text,4,2);
                                                if (StatusDsn1.Text='Dosen Khusus') or
(StatusDsn2.Text='Dosen Khusus')
                                                then Jadual.FieldValues['fixed'] := true
                                                else Jadual.FieldValues['fixed'] := false;
                                            end;
                                            Jadual.post;
                                        end;
                                    Jadual.Next;
                                end;
                            end;
                            Jadual.close;
                            Jadual.open;
                            ClearJ.Click;
                            formproses.bacaulang.caption := '1';
                        end;
                    end;
                end;
            end;
        end;
    end;
    procedure TFormUtama.ViewMKClick(Sender: TObject);
    begin

```

```

formcetak.querymk.open;
formcetak.LaporanMK.preview;
end;

procedure TFormUtama.CariKodeMKClick(Sender: TObject);

begin
    MatKul.First;
    PanelCariMK.show;
    PanelMK.Enabled := false;
    Caller.Caption := 'PanelMK';
end;

procedure TFormUtama.PilihRClick(Sender: TObject);
var
    ketemu : boolean;
begin
    if (ListRSedia.ItemIndex > -1) then
    begin
        Ruang.First; ketemu := false;
        while not ketemu do
        begin
            if
(Ruang.FieldValues['koder']=ListRSedia.Items[ListRSedia.ItemIndex])
            then ketemu := true;
            Ruang.Next;
        end;
        DeleteMK.enabled := false;
        ListRPilih.Items.Add(ListRSedia.Items[ListRSedia.ItemIndex]);
        ListRSedia.Items.Delete(ListRSedia.ItemIndex);
    end
    else showmessage('Anda belum memilih ruang pada kotak sebelah
kiri');
end;

procedure TFormUtama.BatalRClick(Sender: TObject);
var
    ketemu : boolean;
begin
    if (ListRPilih.ItemIndex > -1) then
    begin
        DeleteMK.enabled := false;
        Ruang.First; ketemu := false;
        while not ketemu do
        begin
            if
(Ruang.FieldValues['koder']=ListRPilih.Items[ListRPilih.ItemIndex])
            then ketemu := true;
            Ruang.Next;
        end;
        ListRSedia.Items.Add(ListRPilih.Items[ListRPilih.ItemIndex]);
        ListRPilih.Items.Delete(ListRPilih.ItemIndex);
    end
    else showmessage('Anda belum memilih ruang pada kotak sebelah
kanan');
end;

procedure TFormUtama.CariDsnClick(Sender: TObject);
begin

```

```

H_Dosen.First;
PanelCariDsn.Show;
DBGridCariDsn.show;
PanelDosen.Enabled := false;
Caller2.Caption := 'PanelDosen';
end;

```

```

end.

```

```

unit Konflik;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs,
    StdCtrls, Buttons;

```

```

type

```

```

    TFormKonflik = class(TForm)
        Label1: TLabel;
        NKDs: TEdit;
        OK: TBitBtn;
        Label3: TLabel;
        JumIterasi: TEdit;
        Siap: TLabel;
        procedure OKClick(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

```

```

var

```

```

    FormKonflik: TFormKonflik;

```

```

implementation

```

```

uses utama, Proses;
{$R *.DFM}

```

```

procedure TFormKonflik.OKClick(Sender: TObject);

```

```

var
    valid : boolean;
    i : byte;
begin
    if (NKDs.Text='') or (NKDs.Text='0') then begin
        messagedlg('Nilai konflik harus terisi!',mtinformation,[mbOK],0);
        NKDs.SetFocus;
    end
    else if (JumIterasi.Text='') or (JumIterasi.Text='0') then begin
        messagedlg('Jumlah iterasi harus terisi!',mtinformation,[mbOK],0);
        JumIterasi.SetFocus;
    end
    else begin
        i := 0;
        valid := true;
        repeat

```

```

        inc(i);
        if not(NKDs.Text[i] in['0'..'9']) then valid := false;
until not valid or (i=length(NKDs.Text));
if not valid then
    messagedlg('Nilai konflik harus berisi
angka!',mtinformation,[mbOK],0)
else!begin
    i := 0;
    valid := true;
    repeat
        inc(i);
        if not(JumIterasi.Text[i] in['0'..'9']) then valid := false;
until not valid or (i=length(JumIterasi.Text));
formkonflik.hide;
if (siap.caption='0') then
    begin
        formutama.Enabled := true;
        siap.caption:='1';
    end
else formproses.enabled := true;
end;
end;
end;
end.

```

unit Proses; ✓

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
Grids, ExtCtrls, StdCtrls, Buttons, DBGrids, Db, DBTables, ComCtrls;

type

```

TFormProses = class(TForm)
    StringGrid1: TStringGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    ProgressBar1: TProgressBar;
    ListBox1: TListBox;
    ListBox2: TListBox;
    ListRKul: TListBox;
    GroupBox1: TGroupBox;
    Iterasi: TListBox;
    Button1: TButton;
    GroupBox2: TGroupBox;
    Label3: TLabel;
    Label4: TLabel;
    Edit2: TEdit;
    Edit3: TEdit;
    GroupBox3: TGroupBox;
    Label5: TLabel;

```

```

Label6: TLabel;
Edit1: TEdit;
Edit4: TEdit;
CetakJadual: TButton;
Query2: TQuery;
Reset: TButton;
mulai: TLabel;
bacaulang: TLabel;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormActivate(Sender: TObject);
procedure BacaTabelRuang;
procedure BacaTabelJadual;
procedure BuatArrayPrak;
procedure Initial;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure cari(ke : byte; s : string; var a,b,c : byte);
procedure HitungKonflikDsn;
procedure HitungKonflikSmt;
procedure HitungKonflikTotal;
procedure TaruhMKYgFixed;
procedure RandomMKYgKonflik;
procedure RandomPrakYgKonflik;
procedure AturRuang;
procedure CekIterasiKembar;
procedure TampilkanHasil(ke : byte);
function ada_konflik_dsn(ke,harike,kotakke : byte; tmp2 : string)
: boolean;
    function ada_konflik_smt(ke,harike,kotakke : byte; tmp : byte) :
boolean;
    function ada_konflik_ruang(ke,harike,kotakke : byte; s : string) :
boolean;
    function ada_konflik_jadual(ke,harike : byte; tmp2 : string) :
boolean;
    procedure KonflikPrak(s : string);
    procedure ResetKonflikPrak;
    procedure CetakJadualClick(Sender: TObject);
    procedure IsiHariSenin;
    procedure IsiHariSelasa;
    procedure IsiHariRabu;
    procedure IsiHariKamis;
    procedure IsiHariJumat;
    procedure ResetClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

const
    max      = 10;
    jumkotak = 6;
    jumhari  = 5;

type
    TMatKul2 = record
        kodemk : string[11];
        kodeds : string[5];

```

```

        sks      : byte;
        kotak    : byte;
        smt      : byte;
    end;
    TPrak = record
        kodemk : string[11];
        kodeds : string[5];
        ruang  : string[5];
        kotak  : byte;
        konflik: boolean;
    end;
    TMatKul = record
        kodemk : string[11];
        namamk : string[22];
        kodeds : string[5];
        ruang  : string[5];
        sks    : byte;
        kotak  : byte;
        smt    : byte;
        fixed  : boolean;
        konf_dsn : longint;
    end;
    TKotak = record
        idx      : byte;
        kuliah   : array [1..max] of TMatKul;
    end;
    THari = record
        Kotak : array [1..jumkotak] of TKotak;
    end;
    TJadual = record
        Hari      : array [1..jumhari] of THari;
        konf_tot  : integer;
    end;

```

```

var
    FormProses: TFormProses;
    jumprak,no,no_urut : byte;
    TabuList : array [0..50] of TJadual;
    Kembaran : array [1..50] of byte;
    Prak : array [1..50] of TPrak; //prak dg kotak>1

```

implementation

```

uses Utama,Konflik,Cetak;
{$R *.DFM}

```

```

procedure TFormProses.FormClose(Sender: TObject; var Action: *
TCloseAction);
begin
    FormProses.hide;
    FormUtama.Enabled := true;
end;

```

```

procedure TFormProses.BitBtn2Click(Sender: TObject);
begin
    close;
end;

```

```

procedure TFormProses.BacaTabelRuang;

```

```

begin
  listRKul.items.clear;
  formutama.Ruang.First;
  while not formutama.Ruang.eof do
    begin
      if (copy(formutama.Ruang.fieldbyname('koder').asString,1,1)='B')
or
      (copy(formutama.Ruang.fieldbyname('koder').asString,1,1)='D')
    then
      listRKul.items.add(formutama.Ruang.fieldbyname('koder').asString);
      formutama.Ruang.next;
    end;
  end;
end;

```

```

procedure TFormProses.BacaTabelJadual;      ✓
var
  harike,kotakke,index : byte;
  nemu : boolean;
begin
  panell.show;
  labell.caption := 'Membaca tabel Jadual';
  labell.left := (panell.width-labell.width) div 2;
  panell.refresh;
  formutama.jadual.first;
  TabuList[no].konf_tot := 0;
  for harike := 1 to jumhari do
    for kotakke := 1 to jumkotak do
      TabuList[no].hari[harike].kotak[kotakke].idx := 0;
    while not formutama.jadual.eof do
      begin
        if (formutama.jadual.fieldbyname('ruang').asString<>'') and
          (formutama.jadual.fieldbyname('hari').asString<>'') and
          (formutama.jadual.fieldbyname('jam').asString<>'') and
          (formutama.jadual.fieldbyname('fixed').asString<>'') and
          (formutama.jadual.fieldbyname('kodeds').asString<>'') then
          begin
            progressbar1.position :=
round(formutama.jadual.recno/formutama.jadual.recordcount*100);
            harike := formutama.jadual.fieldbyname('hari').asinteger;
            kotakke := formutama.jadual.fieldbyname('jam').asinteger;
            inc(TabuList[no].hari[harike].kotak[kotakke].idx);
            index := TabuList[no].hari[harike].kotak[kotakke].idx;
            TabuList[no].hari[harike].kotak[kotakke].kuliah[index].kode mk
:= formutama.jadual.fieldbyname('kode mk').asString;

formutama.MatKul.Findkey([formutama.jadual.fieldbyname('kode mk').asstr
ing]);
            TabuList[no].hari[harike].kotak[kotakke].kuliah[index].namamk
:= formutama.MatKul.fieldbyname('namamk').asString;
            TabuList[no].hari[harike].kotak[kotakke].kuliah[index].kodeds
:= formutama.jadual.fieldbyname('kodeds').asString;
            TabuList[no].hari[harike].kotak[kotakke].kuliah[index].ruang
:= formutama.jadual.fieldbyname('ruang').asString;
            formutama.MatKul.First; nemu := false;
            while not formutama.MatKul.Eof and not nemu do
              begin

```

```

        if
        (formutama.MatKul.FieldValues['kode mk']=formutama.jadual.fieldbyname('
kode mk').asstring)
            then nemu := true
        else
            formutama.MatKul.Next;
        end;
        TabuList[no].hari[harike].kotak[kotakke].kuliah[index].sks
:= formutama.MatKul.FieldValues['sks'];
        TabuList[no].hari[harike].kotak[kotakke].kuliah[index].kotak
:= formutama.MatKul.FieldValues['kotak'];
        TabuList[no].hari[harike].kotak[kotakke].kuliah[index].smt
:= formutama.MatKul.FieldValues['smt'];
        TabuList[no].hari[harike].kotak[kotakke].kuliah[index].fixed
:= formutama.jadual.fieldbyname('fixed').asboolean;
        if
not (TabuList[no].hari[harike].kotak[kotakke].kuliah[index].fixed) then
        begin
            if
            (copy(TabuList[no].hari[harike].kotak[kotakke].kuliah[index].kode mk,1,
1)='P') and

            (TabuList[no].hari[harike].kotak[kotakke].kuliah[index].kotak>1) then

KonflikPrak(TabuList[no].hari[harike].kotak[kotakke].kuliah[index].kode
mk)
                else if
            (copy(TabuList[no].hari[harike].kotak[kotakke].kuliah[index].kode mk,1,
1)='M') then

TabuList[no].hari[harike].kotak[kotakke].kuliah[index].konf_dsn :=
1000;
                end
            else
TabuList[no].hari[harike].kotak[kotakke].kuliah[index].konf_dsn := 0;
            end;
            formutama.jadual.next;
        end;
    end; ✓

procedure TFormProses.BuatArrayPrak;//berisi praktikum dg kotak>1 ✓
var
    index : byte;
    nemu : boolean;
begin
    index := 0;
    formutama.matkul.first;
    while not formutama.matkul.eof do
        begin
            if
            (copy(formutama.matkul.fieldbyname('kode mk').asstring,1,1)='P') and
            (formutama.matkul.fieldbyname('kotak').asinteger>1) then
                begin
                    formutama.Jadual.First; nemu := false;
                    while not (formutama.Jadual.EOF) and not (nemu) do
                        begin

```



```

        if
        (formutama.Jadual.FieldName('kodemk').asString=formutama.matkul.fiel
dbyname('kodemk').asString) and
            (formutama.jadual.fieldbyname('ruang').asString<>'')
and
            (formutama.jadual.fieldbyname('hari').asString<>'')
and
            (formutama.jadual.fieldbyname('jam').asString<>'')
and
            (formutama.jadual.fieldbyname('fixed').asString<>'')
and
            (formutama.jadual.fieldbyname('kodeds').asString<>'')
then nemu := true
        else
            formutama.Jadual.Next;

```

```

        end;
    if nemu then
        begin
            inc(index);
            Prak[index].kodemk :=
formutama.matkul.fieldbyname('kodemk').asString;
            Prak[index].kodeds :=
formutama.Jadual.FieldName('kodeds').asString;
            Prak[index].ruang :=
formutama.Jadual.FieldName('ruang').asString;
            Prak[index].kotak :=
formutama.matkul.fieldbyname('kotak').asinteger;
            Prak[index].konflik := true;
        end;
    end;
    formutama.matkul.next;
end;
jumprak := index;
end;

```

```

procedure TFormProses.Initial;  ✓
var
    i : byte;
begin
    mulai.Caption := '1';
    no := 0;
    edit2.text := '0';
    edit3.text := '0';
    edit11.text := '0';
    edit4.text := '0';
    iterasi.items.clear;
    iterasi.items.add('Kondisi awal');
    groupbox1.Enabled := false;
    stringgrid1.Cells[0,0] := 'Waktu';
    stringgrid1.Cells[0,2] := '1-2';
    stringgrid1.Cells[0,7] := '3-4';
    stringgrid1.Cells[0,12] := '5-6';
    stringgrid1.Cells[0,17] := '7-8';
    stringgrid1.Cells[0,22] := '9-10';
    stringgrid1.Cells[0,27] := '11-12';
    stringgrid1.Cells[2,0] := 'Senin';
    stringgrid1.Cells[6,0] := 'Selasa';
    stringgrid1.Cells[10,0] := 'Rabu';

```

```

stringgrid1.Cells[14,0] := 'Kamis';
stringgrid1.Cells[18,0] := 'Jumat';
for i := 1 to jumhari do
begin
    stringgrid1.Cells[(i-1)*4+1,1] := 'Mata Kuliah';
    stringgrid1.Cells[(i-1)*4+2,1] := 'Ruang';
    stringgrid1.Cells[(i-1)*4+3,1] := 'Smt';
    stringgrid1.Cells[(i-1)*4+4,1] := 'Dsn';
end;
end;

procedure TFormProses.FormActivate(Sender: TObject); ✓
begin
    if (mulai.Caption = '0') then Initial;
    formproses.refresh;
    if (bacaulang.caption = '1') then
    begin
        formproses.enabled := false;
        BacaTabelRuang;
        BuatArrayPrak;
        BacaTabelJadual;

        TampilkanHasil(no);
        edit3.text := inttostr(no);
        HitungKonflikDsn;
        HitungKonflikSmt;
        HitungKonflikTotal;
        bacaulang.caption := '0';
        formproses.enabled := true;
    end;
end;

procedure TFormProses.cari(ke : byte; s : string; var a,b,c : byte); ✓
var
    harike,kotakke,i : byte;
begin
    for harike := 1 to jumhari do
        for kotakke := 1 to jumkotak do
            for i := 1 to TabuList[ke].Hari[harike].kotak[kotakke].idx do
                if TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].kode mk =
s then
                    begin
                        a := harike;
                        b := kotakke;
                        c := i;
                        exit;
                    end;
end;

end;

procedure TFormProses.ResetKonflikPrak; ✓
var
    i : byte;
begin
    for i := 1 to jumprak do Prak[i].konflik := false;
end;

procedure TFormProses.KonflikPrak(s : string); ✓
var
    i : byte;

```

```

begin
  for i := 1 to jumprak do
    if (Prak[i].kodemk=s) then
      Prak[i].konflik := true;
    end;
  end;

procedure TFormProses.HitungKonflikDsn; ✓
var
  i, harike, kotakke, idx : byte;
  bentrok : boolean;
  tmp1, tmp2 : string;
begin
  progressbar1.position := 0;
  label1.caption := 'Menghitung konflik dosen';
  label1.left := (panel1.width-label1.width) div 2;

  panel1.show;
  panel1.refresh;
  for harike := 1 to jumhari do
    begin
      progressbar1.position := round(harike/jumhari*100);
      for kotakke := 1 to jumkotak-1 do
        for i := 1 to TabuList[no].Hari[harike].kotak[kotakke].idx-1 do
          begin
            idx := i; bentrok := false;
            repeat
              inc(idx);
              tmp1 :=
TabuList[no].Hari[harike].kotak[kotakke].kuliah[i].kodeds;
              tmp2 :=
TabuList[no].Hari[harike].kotak[kotakke].kuliah[idx].kodeds;
              if ((copy(tmp1,2,1)='S') and (copy(tmp2,2,1)='S') and
(tmp1=tmp2)) then bentrok := true;
              if ((copy(tmp1,2,1)<>'S') and (copy(tmp2,2,1)='S') and
((copy(tmp1,2,2)=copy(tmp2,4,2)) or (copy(tmp1,4,2)=copy(tmp2,4,2))))
then bentrok := true;
              if ((copy(tmp1,2,1)='S') and (copy(tmp2,2,1)<>'S') and
((copy(tmp1,4,2)=copy(tmp2,2,2)) or (copy(tmp1,4,2)=copy(tmp2,4,2))))
then bentrok := true;
              if bentrok then
                begin
                  if
not(TabuList[no].hari[harike].kotak[kotakke].kuliah[i].fixed) then
                    inc(TabuList[no].hari[harike].kotak[kotakke].kuliah[i].konf_dsn, strtoint(formkonflik.NKDs.text))
                  else
                    inc(TabuList[no].hari[harike].kotak[kotakke].kuliah[idx].konf_dsn, strtoint(formkonflik.NKDs.text));
                end;
              until bentrok or
(idx=TabuList[no].Hari[harike].kotak[kotakke].idx);
            end;
          end;
        panel1.hide;
      end;
    end;
  end;
end;

```

```

procedure TFormProses.HitungKonflikSmt; v
var
  i, harike, kotakke, idx : byte;
  bentrok : boolean;
  tmp1, tmp2 : byte;
begin
  progressbar1.position := 0;
  label1.caption := 'Menghitung konflik semester';
  label1.left := (panel1.width-label1.width) div 2;
  panel1.show;
  panel1.refresh;
  for harike := 1 to jumhari do
    begin
      progressbar1.position := round(harike/jumhari*100);
      for kotakke := 1 to jumkotak-1 do
        for i := 1 to TabuList[no].Hari[harike].kotak[kotakke].idx-1 do
          begin
            idx := i; bentrok := false;
            repeat
              inc(idx);
              tmp1 :=
TabuList[no].Hari[harike].kotak[kotakke].kuliah[i].smt;
              tmp2 :=
TabuList[no].Hari[harike].kotak[kotakke].kuliah[idx].smt;
              if (tmp1=tmp2) and
(copy(TabuList[no].Hari[harike].kotak[kotakke].kuliah[i].kode mk,1,1)='
M') and

(copy(TabuList[no].Hari[harike].kotak[kotakke].kuliah[idx].kode mk,1,1)
='M') then bentrok := true;
              if bentrok then
                begin
                  if
not(TabuList[no].hari[harike].kotak[kotakke].kuliah[i].fixed) then

inc(TabuList[no].hari[harike].kotak[kotakke].kuliah[i].konf_dsn, strtoint
(formkonflik.NKDs.text))
                  else if
not(TabuList[no].hari[harike].kotak[kotakke].kuliah[idx].fixed) then

inc(TabuList[no].hari[harike].kotak[kotakke].kuliah[idx].konf_dsn, strtoint
(formkonflik.NKDs.text));
                  end;
                until bentrok or
(idx=TabuList[no].Hari[harike].kotak[kotakke].idx);
              end;
            end;
          panel1.hide;
        end;
      end;
    end;
  end;

```

```

procedure TFormProses.HitungKonflikTotal; v
var
  harike, kotakke, i : byte;
begin
  for harike := 1 to jumhari do
    for kotakke := 1 to jumkotak do
      for i := 1 to TabuList[no].Hari[harike].kotak[kotakke].idx do

```

```

        TabuList[no].konf_tot :=
TabuList[no].konf_tot+TabuList[no].Hari[harike].kotak[kotakke].kuliah[
i].konf_dsn;
        edit2.text := inttostr(TabuList[no].konf_tot);
end;

```

```

function TFormProses.ada_konflik_dsn(ke,harike,kotakke : byte; tmp2 :
string) : boolean;
var
    i : byte;
    bentrok : boolean;
    tmp1 : string;
begin
    bentrok := false;
    for i := 1 to TabuList[ke].Hari[harike].kotak[kotakke].idx do
        begin
            tmp1 := TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].kodeds;
            if ((copy(tmp1,2,1)='S') and (copy(tmp2,2,1)='S') and (tmp1=tmp2))
then bentrok := true;
            if ((copy(tmp1,2,1)<>'S') and (copy(tmp2,2,1)='S') and
((copy(tmp1,2,2)=copy(tmp2,4,2)) or (copy(tmp1,4,2)=copy(tmp2,4,2))))
then bentrok := true;
            if ((copy(tmp1,2,1)='S') and (copy(tmp2,2,1)<>'S') and
((copy(tmp1,4,2)=copy(tmp2,2,2)) or (copy(tmp1,4,2)=copy(tmp2,4,2))))
then bentrok := true;
        end;
    ada_konflik_dsn := bentrok;
end;

```

```

function TFormProses.ada_konflik_smt(ke,harike,kotakke : byte; tmp :
byte) : boolean;
var
    i : byte;
    bentrok : boolean;
begin
    bentrok := false;
    for i := 1 to TabuList[ke].Hari[harike].kotak[kotakke].idx do
        begin
            if
(copy(TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].kode_mk,1,1)='
M') and
            (TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].smt=tmp)
then bentrok := true;
        end;
    ada_konflik_smt := bentrok;
end;

```

```

function TFormProses.ada_konflik_jadual(ke,harike : byte; tmp2 :
string) : boolean;
var
    i,kotakke : byte;
    bentrok : boolean;
    tmp1 : string;
begin
    bentrok := false;
    for kotakke := 1 to 6 do
        for i := 1 to TabuList[ke].Hari[harike].kotak[kotakke].idx do
            begin
                tmp1 := TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].kode_mk;

```

```

    if (tmp1=tmp2) then bentrok := true;
end;
ada_konflik_jadual := bentrok;
end;

function TFormProses.ada_konflik_ruang(ke,harike,kotakke : byte; s : ✓
string) : boolean;
var
    i : byte;
    bentrok : boolean;
begin
    bentrok := false;
    i := 0;
    repeat
        inc(i);
        if (TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].ruang=s)
then bentrok := true;
        until (i=TabuList[ke].Hari[harike].kotak[kotakke].idx) or bentrok;
        ada_konflik_ruang := bentrok;
    end;

procedure TFormProses.TaruhMKYgFixed;v
var
    harike,kotakke,index,i,j : byte;
begin
    //MK yg fixed atau prak dan mk yg tdk konflik
    for harike := 1 to jumhari do
        for kotakke := 1 to jumkotak do
            for i := 1 to TabuList[no-1].Hari[harike].kotak[kotakke].idx do
                begin
                    if ((TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].fixed) and
                        (copy(TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].kode mk,1,1)='M')) or
                        ((TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].fixed) and
                        (copy(TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].kode mk,1,1)='P') and
                        (TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].kotak=1)) or
                        ((copy(TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].kode mk,1,1)='M') and
                        (TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].konf_dsn=0)) or
                        ((copy(TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].kode mk,1,1)='P') and
                        (TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].kotak=1) and
                        (TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].konf_dsn=0)) then
                        begin
                            inc(TabuList[no].Hari[harike].kotak[kotakke].idx,1);
TabuList[no].Hari[harike].kotak[kotakke].kuliah[TabuList[no].Hari[hari
ke].kotak[kotakke].idx] :=
                                TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i];
                                TabuList[no].Hari[harike].kotak[kotakke].kuliah[i].konf_dsn
:= 0;
                        end;

```

```

end;
//praktikum dg kotak>1 yg tdk konflik
for i := 1 to jumprak do
  if not(Prak[i].konflik) then
    begin
      cari(no-1,Prak[i].kodemk,harike,kotakke,index);
      for j := 1 to Prak[i].kotak do
        begin
          if (harike=5) and (kotakke+j-1=3) then inc(kotakke);
          inc(TabuList[no].Hari[harike].kotak[kotakke+j-1].idx,1);
          TabuList[no].Hari[harike].kotak[kotakke+j-1].kuliah[TabuList[no-1].Hari[harike].kotak[kotakke+j-1].idx] :=
            TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[index];
          TabuList[no].Hari[harike].kotak[kotakke+j-1].kuliah[TabuList[no].Hari[harike].kotak[kotakke+j-1].idx].konf_dsn
            := 0;
        end;
      end;
    end;
end;

```

```

procedure TFormProses.RandomMKYgKonflik;
var
  new_hari,new_kotak,harike,kotakke,i : byte;
  valid : boolean;
begin
  randomize;
  progressbar1.position := 0;
  labell.caption := 'Mengacak Mata Kuliah';
  labell.left := (panell.width-labell.width) div 2;
  panell.show;
  panell.refresh;
  for harike := 1 to jumhari do
    for kotakke := 1 to jumkotak do
      begin
        progressbar1.position := round(kotakke/((jumkotak-1)*jumhari)*100);
        for i := 1 to TabuList[no-1].Hari[harike].kotak[kotakke].idx do
          if (copy(TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i].kodemk,1,1)='M') or
            ((copy(TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i].kodemk,1,1)='P') and
              (TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i].kotak=1)) then
            begin
              if (TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i].konf_dsn>0) and
                not (TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i].fixed) then
                begin
                  repeat
                    new_hari := random(jumhari)+1;
                    new_kotak := random(5)+1;
                    if ((new_hari = 5) and (new_kotak = 3)) or
                      (TabuList[no].Hari[new_hari].kotak[new_kotak].idx>=6)
                    or
                      (ada_konflik_dsn(no,new_hari,new_kotak,TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i].kodeds)) or
                      (ada_konflik_smt(no,new_hari,new_kotak,TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i].smt)) or

```

```

        (ada_konflik_jadual(no,new_hari,TabuList[no-
1].Hari[harike].kotak[kotakke].kuliah[i].kode_mk)) then
            valid := false
        else valid := true;
    until valid;
    inc(TabuList[no].Hari[new_hari].kotak[new_kotak].idx,1);

TabuList[no].Hari[new_hari].kotak[new_kotak].kuliah[TabuList[no].Hari[
new_hari].kotak[new_kotak].idx] :=
    TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[i];

TabuList[no].Hari[new_hari].kotak[new_kotak].kuliah[TabuList[no].Hari[
new_hari].kotak[new_kotak].idx].konf_dsn := 0;
end;
end;
end;
panell.hide;
end;

procedure TFormProses.RandomPrakYgKonflik; ✓
var
    index,new_hari,new_kotak,harike,kotakke,i,j,k,tmp : byte;
    valid : boolean;
begin
    progressbar1.position := 0;
    labell.caption := 'Mengacak Praktikum';
    labell.left := (panell.width-labell.width) div 2;
    panell.show;
    panell.refresh;
    randomize;
    for j := 1 to jumprak do
        if Prak[j].konflik then
            begin
                progressbar1.position := round(j/jumprak*100);
                repeat
                    i := 0; valid := true;
                    new_hari := random(jumhari-1)+1;
                    new_kotak := random(6-Prak[j].kotak)+1;
                    tmp := new_kotak;
                    if (new_hari = 5) and (new_kotak = 3) then valid := false;
                    if valid then
                        repeat
                            inc(i);
                            if (new_hari=5) and (tmp+i-1=3) then inc(tmp);
                            if (TabuList[no].Hari[new_hari].kotak[tmp+i-1].idx>=6) or
                                (ada_konflik_dsn(no,new_hari,tmp+i-1,Prak[j].kodeds)) or
                                (ada_konflik_ruang(no,new_hari,tmp+i-1,Prak[j].ruang))
                        until valid := false;
                    until not valid or (i=Prak[j].kotak);
                until valid;
                cari(no-1,Prak[j].kode_mk,harike,kotakke,index);
                for k := 1 to Prak[j].kotak do
                    begin
                        if (new_hari=5) and (new_kotak+k-1=3) then inc(new_kotak);
                        inc(TabuList[no].Hari[new_hari].kotak[new_kotak+k-1].idx,1);
                        TabuList[no].Hari[new_hari].kotak[new_kotak+k-
1].kuliah[TabuList[no].Hari[new_hari].kotak[new_kotak+k-1].idx] :=
                            TabuList[no-1].Hari[harike].kotak[kotakke].kuliah[index];
                    end;
                end;
            end;
        end;
    end;
end;

```



```

        TabuList[no].Hari[new_hari].kotak[new_kotak+k-
1].kuliah[TabuList[no].Hari[new_hari].kotak[new_kotak+k-
1].idx].konf_dsn := 0;
    end;
    Prak[j].konflik := false;
end;
panell1.hide;
end;

```

```

procedure TFormProses.AturRuangan;
var
    harike,kotakke,i,idx1 : byte;
begin
    for harike := 1 to jumhari do
        for kotakke := 1 to jumkotak-1 do
            begin
                idx1 := 0;
                for i := 1 to TabuList[no].Hari[harike].kotak[kotakke].idx do
                    begin
                        if
copy(TabuList[no].Hari[harike].kotak[kotakke].kuliah[i].kode_mk,1,1)='M
' then
                            begin
                                TabuList[no].Hari[harike].kotak[kotakke].kuliah[i].ruang
:= listRKul.Items[idx1];
                                inc(idx1);
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

procedure TFormProses.CekIterasiKembar;
var
    harike,kotakke,nomor,i : byte;
    kembar : boolean;
begin
    nomor := 0; kembaran[no] := 0;
    repeat
        inc(nomor);
        kembar := true;
        harike := 0;
        repeat
            inc(harike);
            kotakke := 0;
            repeat
                inc(kotakke);
                i := 0;
                repeat
                    inc(i);
                    if
(TabuList[nomor].Hari[harike].kotak[kotakke].kuliah[i].kode_mk <>
TabuList[no].Hari[harike].kotak[kotakke].kuliah[i].kode_mk) then
                        kembar := false;
                    until (i=TabuList[nomor].Hari[harike].kotak[kotakke].idx) or
not kembar;
                    until (kotakke=jumkotak) or not kembar;
                    until (harike=jumhari) or not kembar;
                    if kembar then kembaran[no] := nomor;

```

```

until (nomor=no-1) or (kembaran{no}<>0);
end;

```

```

procedure TFormProses.TampilkanHasil(ke : byte); ✓
var
    harike,kotakke,i : byte;
    tmp : string;
begin
    for harike := 1 to stringgrid1.ColCount do
        for kotakke := 2 to stringgrid1.RowCount do
            stringgrid1.Cells[harike,kotakke] := '';
        for harike := 1 to jumhari do
            for kotakke := 1 to jumkotak do
                for i := 1 to TabuList[ke].Hari[harike].kotak[kotakke].idx do
                    begin
                        stringgrid1.Cells[((harike-1)*4)+1,((kotakke-1)*5)+i+1] :=
TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].namamk;
                        stringgrid1.Cells[((harike-1)*4)+2,((kotakke-1)*5)+i+1] :=
TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].ruang;
                        stringgrid1.Cells[((harike-1)*4)+3,((kotakke-1)*5)+i+1] :=
inttostr(TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].smt);
                        tmp :=
TabuList[ke].Hari[harike].kotak[kotakke].kuliah[i].kodeds;
                        if tmp[2] = 'S' then
                            tmp := copy(tmp,4,2)
                        else
                            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
                        stringgrid1.Cells[((harike-1)*4)+4,((kotakke-1)*5)+i+1] :=
tmp;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

procedure TFormProses.BitBtn1Click(Sender: TObject); ✓
var
    harike,kotakke,i,n : byte;
begin
    bitbtn1.enabled := false;
    bitbtn2.enabled := false;
    repeat
        inc(no);
        edit3.text := inttostr(no);
        TabuList[no].konf_tot := 0;
        for harike := 1 to jumhari do
            for kotakke := 1 to jumkotak do
                TabuList[no].hari[harike].kotak[kotakke].idx := 0;
            TaruhMKYgFixed;
            RandomPrakYgKonflik;
            RandomMKYgKonflik;
            AturRuangan;
            if (no>1) then CekIterasiKembar
            else kembaran[1] := 0;
            TampilkanHasil(no);
            HitungKonflikDsn;
            HitungKonflikSmt;
            HitungKonflikTotal;
            if (kembaran[no]=0) then iterasi.items.add('Iterasi ke-
'+inttostr(no))
            else iterasi.items.add('Iterasi ke-'+inttostr(no)+'-
>'+inttostr(kembaran[no]));
        end;
    end;
end;

```

```

until ((no=StrToInt(FormKonflik.JumIterasi.Text)) or ((no>1) and
(TabuList[no].konf_tot=0)));
bitbtn2.Enabled := true;
groupbox1.Enabled := true;
//Memilih TabuList dg nilai konflik terkecil
n := 1;
for i := 2 to no do
    if TabuList[i].konf_tot < TabuList[n].konf_tot then n := i;
iterasi.ItemIndex := n;
edit1.Text := IntToStr(n);
edit4.Text := IntToStr(TabuList[n].konf_tot);
TampilkanHasil(n);
end;

procedure TFormProses.Button1Click(Sender: TObject);
begin
    TampilkanHasil(iterasi.ItemIndex);
    edit3.Text := IntToStr(iterasi.ItemIndex);
    edit2.Text := IntToStr(TabuList[iterasi.ItemIndex].konf_tot);
end;

procedure TFormProses.IsiHariSenin;
var
    i : byte;
    tmp : string;
begin
    for i := 1 to TabuList[iterasi.itemindex].Hari[1].kotak[1].idx do
    begin
        formcetak.MK11.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[1].
        kuliah[i].namamk);

        formcetak.R11.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[1].k
        uliah[i].ruang);

        formcetak.Smt11.Lines.Add(IntToStr(TabuList[iterasi.itemindex].Hari[1]
        .kotak[1].kuliah[i].smt));
        tmp :=
        TabuList[iterasi.itemindex].Hari[1].kotak[1].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp, 4, 2)
        else
            tmp := copy(tmp, 2, 2) + ',' + copy(tmp, 4, 2);
        formcetak.Dsn11.Lines.Add(tmp);
    end;
    for i := 1 to TabuList[iterasi.itemindex].Hari[1].kotak[2].idx do
    begin
        formcetak.MK12.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[2].
        kuliah[i].namamk);

        formcetak.R12.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[2].k
        uliah[i].ruang);

        formcetak.Smt12.Lines.Add(IntToStr(TabuList[iterasi.itemindex].Hari[1]
        .kotak[2].kuliah[i].smt));
        tmp :=
        TabuList[iterasi.itemindex].Hari[1].kotak[2].kuliah[i].kodeds;
        if tmp[2] = 'S' then

```

```

        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
        formcetak.Dsn12.Lines.Add(tmp);
    end;
    for i := 1 to TabuList[iterasi.itemindex].Hari[1].kotak[3].idx do
    begin
        formcetak.MK13.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[3].
        kuliah[i].namamk);

        formcetak.R13.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[3].k
        uliah[i].ruang);

        formcetak.Smt13.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[1]
        .kotak[3].kuliah[i].smt));
        tmp :=
        TabuList[iterasi.itemindex].Hari[1].kotak[3].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn13.Lines.Add(tmp);
        end;
        for i := 1 to TabuList[iterasi.itemindex].Hari[1].kotak[4].idx do
        begin

            formcetak.MK14.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[4].
            kuliah[i].namamk);

            formcetak.R14.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[4].k
            uliah[i].ruang);

            formcetak.Smt14.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[1]
            .kotak[4].kuliah[i].smt));
            tmp :=
            TabuList[iterasi.itemindex].Hari[1].kotak[4].kuliah[i].kodeds;
            if tmp[2] = 'S' then
                tmp := copy(tmp,4,2)
            else
                tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
                formcetak.Dsn14.Lines.Add(tmp);
            end;
            for i := 1 to TabuList[iterasi.itemindex].Hari[1].kotak[5].idx do
            begin

                formcetak.MK15.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[5].
                kuliah[i].namamk);

                formcetak.R15.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[5].k
                uliah[i].ruang);

                formcetak.Smt15.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[1]
                .kotak[5].kuliah[i].smt));
                tmp :=
                TabuList[iterasi.itemindex].Hari[1].kotak[5].kuliah[i].kodeds;
                if tmp[2] = 'S' then
                    tmp := copy(tmp,4,2)
                else

```

```

        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
        formcetak.Dsn15.Lines.Add(tmp);
    end;
    for i := 1 to TabuList[iterasi.itemindex].Hari[1].kotak[6].idx do
    begin

        formcetak.MK16.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[6].
        kuliah[i].namamk);

        formcetak.R16.Lines.Add(TabuList[iterasi.itemindex].Hari[1].kotak[6].k
        uliah[i].ruang);

        formcetak.Smt16.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[1]
        .kotak[6].kuliah[i].smt));
        tmp :=
        TabuList[iterasi.itemindex].Hari[1].kotak[6].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn16.Lines.Add(tmp);
        end;
    end;
    ...

procedure TFormProses.IsiHariSelasa;
var
    i : byte;
    tmp : string;
begin
    for i := 1 to TabuList[iterasi.itemindex].Hari[2].kotak[1].idx do
    begin

        formcetak.MK21.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[1].
        kuliah[i].namamk);

        formcetak.R21.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[1].k
        uliah[i].ruang);

        formcetak.Smt21.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[2]
        .kotak[1].kuliah[i].smt));
        tmp :=
        TabuList[iterasi.itemindex].Hari[2].kotak[1].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn21.Lines.Add(tmp);
        end;
        for i := 1 to TabuList[iterasi.itemindex].Hari[2].kotak[2].idx do
        begin

            formcetak.MK22.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[2].
            kuliah[i].namamk);

            formcetak.R22.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[2].k
            uliah[i].ruang);

            formcetak.Smt22.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[2]
            .kotak[2].kuliah[i].smt));

```

```

    tmp :=
TabuList[iterasi.itemindex].Hari[2].kotak[2].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn22.Lines.Add(tmp);
end;
for i := 1 to TabuList[iterasi.itemindex].Hari[2].kotak[3].idx do
begin

formcetak.MK23.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[3].
kuliah[i].namamk);

formcetak.R23.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[3].k
uliah[i].ruang);

formcetak.Smt23.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[2]
.kotak[3].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[2].kotak[3].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn23.Lines.Add(tmp);
end;
for i := 1 to TabuList[iterasi.itemindex].Hari[2].kotak[4].idx do
begin

formcetak.MK24.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[4].
kuliah[i].namamk);

formcetak.R24.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[4].k
uliah[i].ruang);

formcetak.Smt24.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[2]
.kotak[4].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[2].kotak[4].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn24.Lines.Add(tmp);
end;
for i := 1 to TabuList[iterasi.itemindex].Hari[2].kotak[5].idx do
begin

formcetak.MK25.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[5].
kuliah[i].namamk);

formcetak.R25.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[5].k
uliah[i].ruang);

formcetak.Smt25.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[2]
.kotak[5].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[2].kotak[5].kuliah[i].kodeds;

```

```

    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
        formcetak.Dsn25.Lines.Add(tmp);
    end;
    for i := 1 to TabuList[iterasi.itemindex].Hari[2].kotak[6].idx do
        begin

formcetak.MK26.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[6].
kuliah[i].namamk);

formcetak.R26.Lines.Add(TabuList[iterasi.itemindex].Hari[2].kotak[6].k
uliah[i].ruang);

formcetak.Smt26.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[2]
.kotak[6].kuliah[i].smt));
        tmp :=
TabuList[iterasi.itemindex].Hari[2].kotak[6].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn26.Lines.Add(tmp);
        end;
    end;

procedure TFormProses.IsiHariRabu;
var
    i : byte;
    tmp : string;
begin
    for i := 1 to TabuList[iterasi.itemindex].Hari[3].kotak[1].idx do
        begin

formcetak.MK31.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[1].
kuliah[i].namamk);

formcetak.R31.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[1].k
uliah[i].ruang);

formcetak.Smt31.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[3]
.kotak[1].kuliah[i].smt));
        tmp :=
TabuList[iterasi.itemindex].Hari[3].kotak[1].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn31.Lines.Add(tmp);
        end;
    for i := 1 to TabuList[iterasi.itemindex].Hari[3].kotak[2].idx do
        begin

formcetak.MK32.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[2].
kuliah[i].namamk);

formcetak.R32.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[2].k
uliah[i].ruang);

```

```

formcetak.Smt32.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[3]
.kotak[2].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[3].kotak[2].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn32.Lines.Add(tmp);
end;
for i := 1 to TabuList[iterasi.itemindex].Hari[3].kotak[3].idx do
begin

```

```

formcetak.MK33.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[3].
kuliah[i].namamk);

```

```

formcetak.R33.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[3].k
uliah[i].ruang);

```

```

formcetak.Smt33.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[3]
.kotak[3].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[3].kotak[3].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn33.Lines.Add(tmp);
end;
for i := 1 to TabuList[iterasi.itemindex].Hari[3].kotak[4].idx do
begin

```

```

formcetak.MK34.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[4].
kuliah[i].namamk);

```

```

formcetak.R34.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[4].k
uliah[i].ruang);

```

```

formcetak.Smt34.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[3]
.kotak[4].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[3].kotak[4].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn34.Lines.Add(tmp);
end;
for i := 1 to TabuList[iterasi.itemindex].Hari[3].kotak[5].idx do
begin

```

```

formcetak.MK35.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[5].
kuliah[i].namamk);

```

```

formcetak.R35.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[5].k
uliah[i].ruang);

```



```

formcetak.Smt35.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[3]
.kotak[5].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[3].kotak[5].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn35.Lines.Add(tmp);
end;
for i := 1 to TabuList[iterasi.itemindex].Hari[3].kotak[6].idx do
begin

```

```

formcetak.MK36.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[6].
kuliah[i].namamk);

```

```

formcetak.R36.Lines.Add(TabuList[iterasi.itemindex].Hari[3].kotak[6].k
uliah[i].ruang);

```

```

formcetak.Smt36.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[3]
.kotak[6].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[3].kotak[6].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn36.Lines.Add(tmp);
end;
end;

```

```

procedure TFormProses.IsiHariKamis;

```

```

var
    i : byte;
    tmp : string;
begin
    for i := 1 to TabuList[iterasi.itemindex].Hari[4].kotak[1].idx do
    begin

```

```

formcetak.MK41.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[1].
kuliah[i].namamk);

```

```

formcetak.R41.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[1].k
uliah[i].ruang);

```

```

formcetak.Smt41.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[4]
.kotak[1].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[4].kotak[1].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
    formcetak.Dsn41.Lines.Add(tmp);

```

```

end;
for i := 1 to TabuList[iterasi.itemindex].Hari[4].kotak[2].idx do
begin

```

```
formcetak.MK42.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[2].  
kuliah[i].namamk);
```

```
formcetak.R42.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[2].k  
uliah[i].ruang);
```

```
formcetak.Smt42.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[4]  
.kotak[2].kuliah[i].smt));
```

```
    tmp :=  
TabuList[iterasi.itemindex].Hari[4].kotak[2].kuliah[i].kodeds;  
    if tmp[2] = 'S' then  
        tmp := copy(tmp,4,2)  
    else  
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);  
        formcetak.Dsn42.Lines.Add(tmp);  
    end;  
    for i := 1 to TabuList[iterasi.itemindex].Hari[4].kotak[3].idx do  
begin
```

```
formcetak.MK43.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[3].  
kuliah[i].namamk);
```

```
formcetak.R43.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[3].k  
uliah[i].ruang);
```

```
formcetak.Smt43.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[4]  
.kotak[3].kuliah[i].smt));
```

```
    tmp :=  
TabuList[iterasi.itemindex].Hari[4].kotak[3].kuliah[i].kodeds;  
    if tmp[2] = 'S' then  
        tmp := copy(tmp,4,2)  
    else  
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);  
        formcetak.Dsn43.Lines.Add(tmp);  
    end;  
    for i := 1 to TabuList[iterasi.itemindex].Hari[4].kotak[4].idx do  
begin
```

```
formcetak.MK44.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[4].  
kuliah[i].namamk);
```

```
formcetak.R44.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[4].k  
uliah[i].ruang);
```

```
formcetak.Smt44.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[4]  
.kotak[4].kuliah[i].smt));
```

```
    tmp :=  
TabuList[iterasi.itemindex].Hari[4].kotak[4].kuliah[i].kodeds;  
    if tmp[2] = 'S' then  
        tmp := copy(tmp,4,2)  
    else  
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);  
        formcetak.Dsn44.Lines.Add(tmp);  
    end;  
    for i := 1 to TabuList[iterasi.itemindex].Hari[4].kotak[5].idx do  
begin
```

```
formcetak.MK45.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[5].  
kuliah[i].namamk);
```

```
formcetak.R45.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[5].k  
uliah[i].ruang);
```

```
formcetak.Smt45.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[4]  
.kotak[5].kuliah[i].smt));
```

```
tmp :=  
TabuList[iterasi.itemindex].Hari[4].kotak[5].kuliah[i].kodeds;
```

```
if tmp[2] = 'S' then
```

```
tmp := copy(tmp,4,2)
```

```
else
```

```
tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
```

```
formcetak.Dsn45.Lines.Add(tmp);
```

```
end;
```

```
for i := 1 to TabuList[iterasi.itemindex].Hari[4].kotak[6].idx do
```

```
begin
```

```
formcetak.MK46.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[6].  
kuliah[i].namamk);
```

```
formcetak.R46.Lines.Add(TabuList[iterasi.itemindex].Hari[4].kotak[6].k  
uliah[i].ruang);
```

```
formcetak.Smt46.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[4]  
.kotak[6].kuliah[i].smt));
```

```
tmp :=
```

```
TabuList[iterasi.itemindex].Hari[4].kotak[6].kuliah[i].kodeds;
```

```
if tmp[2] = 'S' then
```

```
tmp := copy(tmp,4,2)
```

```
else
```

```
tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
```

```
formcetak.Dsn46.Lines.Add(tmp);
```

```
end;
```

```
end;
```

```
procedure TFormProses.IsiHariJumat;
```

```
var
```

```
i : byte;
```

```
tmp : string;
```

```
begin
```

```
for i := 1 to TabuList[iterasi.itemindex].Hari[5].kotak[1].idx do
```

```
begin
```

```
formcetak.MK51.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[1].  
kuliah[i].namamk);
```

```
formcetak.R51.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[1].k  
uliah[i].ruang);
```

```
formcetak.Smt51.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[5]  
.kotak[1].kuliah[i].smt));
```

```
tmp :=
```

```
TabuList[iterasi.itemindex].Hari[5].kotak[1].kuliah[i].kodeds;
```

```
if tmp[2] = 'S' then
```

```
tmp := copy(tmp,4,2)
```

```
else
```

```

        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
        formcetak.Dsn51.Lines.Add(tmp);
    end;
    for i := 1 to TabuList[iterasi.itemindex].Hari[5].kotak[2].idx do
    begin

formcetak.MK52.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[2].
kuliah[i].namamk);

        formcetak.R52.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[2].k
uliah[i].ruang);

        formcetak.Smt52.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[5]
.kotak[2].kuliah[i].smt));
        tmp :=
TabuList[iterasi.itemindex].Hari[5].kotak[2].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn52.Lines.Add(tmp);
        end;
        for i := 1 to TabuList[iterasi.itemindex].Hari[5].kotak[3].idx do
        begin

formcetak.MK53.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[3].
kuliah[i].namamk);

        formcetak.R53.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[3].k
uliah[i].ruang);

        formcetak.Smt53.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[5]
.kotak[3].kuliah[i].smt));
        tmp :=
TabuList[iterasi.itemindex].Hari[5].kotak[3].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn53.Lines.Add(tmp);
        end;
        for i := 1 to TabuList[iterasi.itemindex].Hari[5].kotak[4].idx do
        begin

formcetak.MK54.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[4].
kuliah[i].namamk);

        formcetak.R54.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[4].k
uliah[i].ruang);

        formcetak.Smt54.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[5]
.kotak[4].kuliah[i].smt));
        tmp :=
TabuList[iterasi.itemindex].Hari[5].kotak[4].kuliah[i].kodeds;
        if tmp[2] = 'S' then
            tmp := copy(tmp,4,2)
        else
            tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
            formcetak.Dsn54.Lines.Add(tmp);

```

```

end;
for i := 1 to TabuList[iterasi.itemindex].Hari[5].kotak[5].idx do
begin
formcetak.MK55.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[5].
kuliah[i].namamk);

formcetak.R55.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[5].k
uliah[i].ruang);

formcetak.Smt55.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[5]
.kotak[5].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[5].kotak[5].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
        formcetak.Dsn55.Lines.Add(tmp);
    end;
for i := 1 to TabuList[iterasi.itemindex].Hari[5].kotak[6].idx do
begin,

formcetak.MK56.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[6].
kuliah[i].namamk);

formcetak.R56.Lines.Add(TabuList[iterasi.itemindex].Hari[5].kotak[6].k
uliah[i].ruang);

formcetak.Smt56.Lines.Add(inttostr(TabuList[iterasi.itemindex].Hari[5]
.kotak[6].kuliah[i].smt));
    tmp :=
TabuList[iterasi.itemindex].Hari[5].kotak[6].kuliah[i].kodeds;
    if tmp[2] = 'S' then
        tmp := copy(tmp,4,2)
    else
        tmp := copy(tmp,2,2) + ',' + copy(tmp,4,2);
        formcetak.Dsn56.Lines.Add(tmp);
    end;
end;

procedure TFormProses.CetakJadualClick(Sender: TObject);
begin
    //Hapus Hari Senin
    formcetak.MK11.Lines.Clear;formcetak.R11.Lines.Clear;
    formcetak.MK12.Lines.Clear;formcetak.R12.Lines.Clear;
    formcetak.MK13.Lines.Clear;formcetak.R13.Lines.Clear;
    formcetak.MK14.Lines.Clear;formcetak.R14.Lines.Clear;
    formcetak.MK15.Lines.Clear;formcetak.R15.Lines.Clear;
    formcetak.MK16.Lines.Clear;formcetak.R16.Lines.Clear;
    formcetak.Smt11.Lines.Clear;formcetak.Dsn11.Lines.Clear;
    formcetak.Smt12.Lines.Clear;formcetak.Dsn12.Lines.Clear;
    formcetak.Smt13.Lines.Clear;formcetak.Dsn13.Lines.Clear;
    formcetak.Smt14.Lines.Clear;formcetak.Dsn14.Lines.Clear;
    formcetak.Smt15.Lines.Clear;formcetak.Dsn15.Lines.Clear;
    formcetak.Smt16.Lines.Clear;formcetak.Dsn16.Lines.Clear;
    //Hapus Hari Selasa
    formcetak.MK21.Lines.Clear;formcetak.R21.Lines.Clear;
    formcetak.MK22.Lines.Clear;formcetak.R22.Lines.Clear;

```

```

formcetak.MK23.Lines.Clear;formcetak.R23.Lines.Clear;
formcetak.MK24.Lines.Clear;formcetak.R24.Lines.Clear;
formcetak.MK25.Lines.Clear;formcetak.R25.Lines.Clear;
formcetak.MK26.Lines.Clear;formcetak.R26.Lines.Clear;
formcetak.Smt21.Lines.Clear;formcetak.Dsn21.Lines.Clear;
formcetak.Smt22.Lines.Clear;formcetak.Dsn22.Lines.Clear;
formcetak.Smt23.Lines.Clear;formcetak.Dsn23.Lines.Clear;
formcetak.Smt24.Lines.Clear;formcetak.Dsn24.Lines.Clear;
formcetak.Smt25.Lines.Clear;formcetak.Dsn25.Lines.Clear;
formcetak.Smt26.Lines.Clear;formcetak.Dsn26.Lines.Clear;
//Hapus Hari Rabu
formcetak.MK31.Lines.Clear;formcetak.R31.Lines.Clear;
formcetak.MK32.Lines.Clear;formcetak.R32.Lines.Clear;
formcetak.MK33.Lines.Clear;formcetak.R33.Lines.Clear;
formcetak.MK34.Lines.Clear;formcetak.R34.Lines.Clear;
formcetak.MK35.Lines.Clear;formcetak.R35.Lines.Clear;
formcetak.MK36.Lines.Clear;formcetak.R36.Lines.Clear;
formcetak.Smt31.Lines.Clear;formcetak.Dsn31.Lines.Clear;
formcetak.Smt32.Lines.Clear;formcetak.Dsn32.Lines.Clear;
formcetak.Smt33.Lines.Clear;formcetak.Dsn33.Lines.Clear;
formcetak.Smt34.Lines.Clear;formcetak.Dsn34.Lines.Clear;
formcetak.Smt35.Lines.Clear;formcetak.Dsn35.Lines.Clear;
formcetak.Smt36.Lines.Clear;formcetak.Dsn36.Lines.Clear;
//Hapus Hari Kamis
formcetak.MK41.Lines.Clear;formcetak.R41.Lines.Clear;
formcetak.MK42.Lines.Clear;formcetak.R42.Lines.Clear;
formcetak.MK43.Lines.Clear;formcetak.R43.Lines.Clear;
formcetak.MK44.Lines.Clear;formcetak.R44.Lines.Clear;
formcetak.MK45.Lines.Clear;formcetak.R45.Lines.Clear;
formcetak.MK46.Lines.Clear;formcetak.R46.Lines.Clear;
formcetak.Smt41.Lines.Clear;formcetak.Dsn41.Lines.Clear;
formcetak.Smt42.Lines.Clear;formcetak.Dsn42.Lines.Clear;
formcetak.Smt43.Lines.Clear;formcetak.Dsn43.Lines.Clear;
formcetak.Smt44.Lines.Clear;formcetak.Dsn44.Lines.Clear;
formcetak.Smt45.Lines.Clear;formcetak.Dsn45.Lines.Clear;
formcetak.Smt46.Lines.Clear;formcetak.Dsn46.Lines.Clear;
//Hapus Hari Jumat
formcetak.MK51.Lines.Clear;formcetak.R51.Lines.Clear;
formcetak.MK52.Lines.Clear;formcetak.R52.Lines.Clear;
formcetak.MK53.Lines.Clear;formcetak.R53.Lines.Clear;
formcetak.MK54.Lines.Clear;formcetak.R54.Lines.Clear;
formcetak.MK55.Lines.Clear;formcetak.R55.Lines.Clear;
formcetak.MK56.Lines.Clear;formcetak.R56.Lines.Clear;
formcetak.Smt51.Lines.Clear;formcetak.Dsn51.Lines.Clear;
formcetak.Smt52.Lines.Clear;formcetak.Dsn52.Lines.Clear;
formcetak.Smt53.Lines.Clear;formcetak.Dsn53.Lines.Clear;
formcetak.Smt54.Lines.Clear;formcetak.Dsn54.Lines.Clear;
formcetak.Smt55.Lines.Clear;formcetak.Dsn55.Lines.Clear;
formcetak.Smt56.Lines.Clear;formcetak.Dsn56.Lines.Clear;

//Mempersiapkan tampilan laporan
IsiHariSenin;
IsiHariSelasa;
IsiHariRabu;
IsiHariKamis;
IsiHariJumat;
if (kembaran[iterasi.itemindex]=0) then
    formcetak.Nomor.Caption := inttostr(iterasi.itemindex)
else formcetak.Nomor.Caption := inttostr(iterasi.itemindex)+

```

' (sama dengan iterasi ke-' +

```
inttostr(kembaran[iterasi.itemindex]+'')';
    formcetak.Konflik.Caption :=
inttostr(TabuList[iterasi.ItemIndex].konf_tot);
    formcetak.LaporanJadual.Preview;
end;

procedure TFormProses.ResetClick(Sender: TObject);
begin
    formproses.enabled := false;
    mulai.Caption := '0';
    bitbtn1.Enabled := true;
    formkonflik.Show;
end;

end.
```

unit Cetak;

interface

```
uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
    Dialogs,
    QuickRpt, QrCtrls, ExtCtrls, Db, DBTables;
```

type

```
TFormCetak = class(TForm)
    LaporanJadual: TQuickRep;
    QRBand1: TQRBand;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRShape1: TQRShape;
    QRShape2: TQRShape;
    QRShape3: TQRShape;
    QRShape4: TQRShape;
    QRShape6: TQRShape;
    QRLabel13: TQRLabel;
    QRLabel14: TQRLabel;
    QRLabel15: TQRLabel;
    QRShape11: TQRShape;
    QRShape12: TQRShape;
    QRLabel16: TQRLabel;
    QRShape13: TQRShape;
    QRShape14: TQRShape;
    QRShape15: TQRShape;
    QRLabel17: TQRLabel;
    QRLabel18: TQRLabel;
    QRLabel19: TQRLabel;
    MK11: TQRRichText;
    QRLabel10: TQRLabel;
    QRLabel11: TQRLabel;
    QRLabel12: TQRLabel;
    QRLabel13: TQRLabel;
    QRLabel14: TQRLabel;
    Dsn11: TQRRichText;
    R11: TQRRichText;
```

Smt11: TQRRichText;
QRShape5: TQRShape;
MK12: TQRRichText;
QRShape16: TQRShape;
R12: TQRRichText;
Smt12: TQRRichText;
Dsn12: TQRRichText;
QRShape17: TQRShape;
MK13: TQRRichText;
QRShape25: TQRShape;
R13: TQRRichText;
Smt13: TQRRichText;
Dsn13: TQRRichText;
QRShape18: TQRShape;
MK14: TQRRichText;
QRShape19: TQRShape;
R14: TQRRichText;
Smt14: TQRRichText;
Dsn14: TQRRichText;
QRShape20: TQRShape;
MK15: TQRRichText;
QRShape26: TQRShape;
R15: TQRRichText;
Smt15: TQRRichText;
Dsn15: TQRRichText;
QRShape27: TQRShape;
MK16: TQRRichText;
QRShape28: TQRShape;
R16: TQRRichText;
Smt16: TQRRichText;
Dsn16: TQRRichText;
QRShape7: TQRShape;
QRShape8: TQRShape;
QRShape9: TQRShape;
QRLabel15: TQRLabel;
QRShape10: TQRShape;
QRLabel16: TQRLabel;
QRShape21: TQRShape;
QRLabel17: TQRLabel;
QRLabel18: TQRLabel;
QRLabel19: TQRLabel;
QRShape22: TQRShape;
QRShape23: TQRShape;
QRShape24: TQRShape;
QRLabel20: TQRLabel;
QRShape29: TQRShape;
QRLabel21: TQRLabel;
QRShape30: TQRShape;
QRLabel22: TQRLabel;
QRLabel23: TQRLabel;
QRLabel24: TQRLabel;
QRShape31: TQRShape;
QRShape32: TQRShape;
QRShape33: TQRShape;
QRLabel25: TQRLabel;
QRShape34: TQRShape;
QRLabel26: TQRLabel;
QRShape35: TQRShape;
QRLabel27: TQRLabel;

QRLabel128: TQRLabel;
QRLabel129: TQRLabel;
QRShape36: TQRShape;
QRShape37: TQRShape;
QRShape38: TQRShape;
QRLabel130: TQRLabel;
QRShape39: TQRShape;
QRLabel131: TQRLabel;
QRShape40: TQRShape;
QRLabel132: TQRLabel;
QRLabel133: TQRLabel;
QRLabel134: TQRLabel;
QRShape42: TQRShape;
MK31: TQRRichText;
R31: TQRRichText;
Smt31: TQRRichText;
Dsn31: TQRRichText;
QRShape43: TQRShape;
MK41: TQRRichText;
R41: TQRRichText;
Smt41: TQRRichText;
Dsn41: TQRRichText;
QRShape44: TQRShape;
MK51: TQRRichText;
R51: TQRRichText;
Smt51: TQRRichText;
Dsn51: TQRRichText;
QRShape45: TQRShape;
MK22: TQRRichText;
R22: TQRRichText;
Smt22: TQRRichText;
Dsn22: TQRRichText;
QRShape46: TQRShape;
MK23: TQRRichText;
R23: TQRRichText;
Smt23: TQRRichText;
Dsn23: TQRRichText;
QRShape47: TQRShape;
MK24: TQRRichText;
R24: TQRRichText;
Smt24: TQRRichText;
Dsn24: TQRRichText;
QRShape49: TQRShape;
MK26: TQRRichText;
R26: TQRRichText;
Smt26: TQRRichText;
Dsn26: TQRRichText;
QRShape50: TQRShape;
MK32: TQRRichText;
R32: TQRRichText;
Smt32: TQRRichText;
Dsn32: TQRRichText;
QRShape52: TQRShape;
MK33: TQRRichText;
R33: TQRRichText;
Smt33: TQRRichText;
Dsn33: TQRRichText;
QRShape53: TQRShape;
MK34: TQRRichText;

R34: TQRRichText;
Smt34: TQRRichText;
Dsn34: TQRRichText;
QRShape56: TQRShape;
MK43: TQRRichText;
R43: TQRRichText;
Smt43: TQRRichText;
Dsn43: TQRRichText;
QRShape57: TQRShape;
MK44: TQRRichText;
R44: TQRRichText;
Smt44: TQRRichText;
Dsn44: TQRRichText;
QRShape58: TQRShape;
MK52: TQRRichText;
R52: TQRRichText;
Smt52: TQRRichText;
Dsn52: TQRRichText;
QRShape59: TQRShape;
MK53: TQRRichText;
R53: TQRRichText;
Smt53: TQRRichText;
Dsn53: TQRRichText;
QRShape60: TQRShape;
MK54: TQRRichText;
R54: TQRRichText;
Smt54: TQRRichText;
Dsn54: TQRRichText;
QRShape61: TQRShape;
MK45: TQRRichText;
R45: TQRRichText;
Smt45: TQRRichText;
Dsn45: TQRRichText;
QRShape62: TQRShape;
MK55: TQRRichText;
R55: TQRRichText;
Smt55: TQRRichText;
Dsn55: TQRRichText;
QRShape64: TQRShape;
MK56: TQRRichText;
R56: TQRRichText;
Smt56: TQRRichText;
Dsn56: TQRRichText;
QRShape41: TQRShape;
MK21: TQRRichText;
R21: TQRRichText;
Smt21: TQRRichText;
Dsn21: TQRRichText;
QRShape51: TQRShape;
MK42: TQRRichText;
R42: TQRRichText;
Smt42: TQRRichText;
Dsn42: TQRRichText;
QRShape48: TQRShape;
MK46: TQRRichText;
R46: TQRRichText;
Smt46: TQRRichText;
Dsn46: TQRRichText;
QRShape54: TQRShape;

MK35: TQRRichText;
R35: TQRRichText;
Smt35: TQRRichText;
Dsn35: TQRRichText;
QRShape55: TQRShape;
MK36: TQRRichText;
R36: TQRRichText;
Smt36: TQRRichText;
Dsn36: TQRRichText;
QRShape63: TQRShape;
MK25: TQRRichText;
R25: TQRRichText;
Smt25: TQRRichText;
Dsn25: TQRRichText;
QRLabel35: TQRLabel;
QRLabel36: TQRLabel;
Nomor: TQRLabel;
Konflik: TQRLabel;
QRLabel53: TQRLabel;
QRLabel57: TQRLabel;
QueryRep: TQuery;
QueryDs: TQuery;
DataSource1: TDataSource;
LaporanDsn: TQuickRep;
QRBand4: TQRBand;
QRLabel58: TQRLabel;
DetailBand: TQRBand;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
QRLabel59: TQRLabel;
QRLabel60: TQRLabel;
QRLabel63: TQRLabel;
QRLabel64: TQRLabel;
QRSubDetail1: TQRSubDetail;
QRDBText4: TQRDBText;
QRExp1: TQRExp;
QRGroup1: TQRGroup;
QRLabel65: TQRLabel;
QRLabel66: TQRLabel;
QRBand5: TQRBand;
QRBand6: TQRBand;
QRSysData1: TQRSysData;
QRSysData2: TQRSysData;
LaporanMK: TQuickRep;
QRBand2: TQRBand;
QRLabel37: TQRLabel;
QRBand3: TQRBand;
QRDBText5: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRDBText8: TQRDBText;
QRDBText9: TQRDBText;
QueryMK: TQuery;
QRBand7: TQRBand;
QRLabel38: TQRLabel;
QRLabel39: TQRLabel;
QRLabel40: TQRLabel;
QRLabel41: TQRLabel;
QRLabel42: TQRLabel;

```

    QRLabel43: TQRLabel;
    QRExpr2: TQRExpr;
    QRBand8: TQRBand;
    QRSysData3: TQRSysData;
    QRSysData4: TQRSysData;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure LaporanMKAfterPreview(Sender: TObject);
    procedure LaporanDsnAfterPreview(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    FormCetak: TFormCetak;

implementation

{$R *.DFM}

uses utama;

procedure TFormCetak.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
    FormCetak.Hide;
end;

procedure TFormCetak.LaporanMKAfterPreview(Sender: TObject);
begin
    QueryMK.Close;
end;

procedure TFormCetak.LaporanDsnAfterPreview(Sender: TObject);
begin
    QueryRep.Close;
    QueryDs.Close;
end;

end.

```